JPL Publication 92-32, Vol. II

An Improved Approach for Flight Readiness Certification—Probabilistic Models for Flaw Propagation and Turbine Blade Failure

Volume II: Software Documentation

N. R. Moore
D. H. Ebbeler
L. E. Newlin
S. Sutharshana
M. Creager

December 30, 1992

NASA

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Preface

This report presents a methodology for managing failure risk cost-effectively and evaluating flight readiness of such aerospace systems as launch vehicles and planetary spacecraft. The methodology was developed by the Jet Propulsion Laboratory (JPL) under NASA RTOP 553-02-01 sponsored by the Office of Space Flight (OSF), NASA Headquarters. This work was performed as a part of the Certification Process Assessment task initiated by OSF due to concern about criteria for certifying flight readiness of the Space Shuttle propulsion system. The methodology is not only applicable to flight readiness evaluation, but also to design definition and to the identification of risk control measures during the design, development, or operational phases of a project.

An early phase of this work included an extensive review of certification and failure risk assessment approaches used by the aerospace industry and government agencies. Based on the findings of this review,[1] further work was focused on defining, developing, and demonstrating an improved technical approach for failure risk assessment that can incorporate information from both test experience and analytical modeling to obtain a quantitative failure risk estimate. This approach, called Probabilistic Failure Assessment (PFA), is of particular value when information relevant to failure prediction, including test experience and knowledge of parameters used in analytical modeling of failure phenomena, is expensive or difficult to acquire. Under such constraints, a quantitative evaluation of failure risk based on the information available from both analytical modeling and operating experience is needed to make effective risk management decisions that utilize financial resources efficiently.

The PFA methodology is applicable to failure modes that can be characterized by analytical or empirical modeling of failure phenomena, including those of structural, electro-optical, propulsion, power, and thermal control systems, and is especially useful when models or information used in analysis are uncertain or approximate. PFA can be applied at any time in the design, development, or operational phases of a program to quantitatively estimate failure risk based on the information available at the time of the risk assessment and can be used to evaluate and rank alternative measures to control risk, thereby enabling the more effective allocation of limited financial resources.

---

1 See [14] of Section 1.0 references.

The work documented in this report was carried out by a multidisciplinary team of JPL technical personnel, which was managed by N. R. Moore. This team was composed of individuals with expertise in statistics, systems modeling, and engineering analysis. D. H. Ebbeler formulated and structured the statistical methodology and directed its implementation. L. E. Newlin formulated and implemented probabilistic engineering models and implemented the statistical methodology. S. Sutharshana formulated and implemented probabilistic analytical methods and models. M. Creager[2] made major contributions to defining and formulating the probabilistic modeling approach and analytical modeling procedures used in this work. D. Goode typeset the manuscript, including graphics, using computerized desktop publishing methods, and E. Reinig edited the manuscript.

In developing the PFA methodology, the JPL team interacted with aerospace system manufacturers, the Marshall Space Flight Center, and the Lewis Research Center. Individuals of these organizations generously shared information and spent significant amounts of time with the JPL team. In particular, Rocketdyne, Canoga Park, California; Aerojet TechSystems, Nimbus, California; and Pratt & Whitney, West Palm Beach, Florida, collaborated in performing the application examples given herein. In addition, technical comments on certification approaches and failure modeling were provided by personnel from the above-listed organizations and General Electric, Cincinnati, Ohio; the Federal Aviation Administration; and the Wright-Patterson Air Force Base.

The PFA methodology, examples of its application to spaceflight components, and computer software used to implement PFA are documented in the two volumes of this report. Volume I documents the PFA methodology and the application examples, including the rationale for PFA and the analysis procedures used in the examples. Volume II contains detailed documentation of the computer software used to implement PFA for the application examples, including user's guides, code execution examples, flowcharts, and listings of the computer programs.

---

[2] Currently of Structural Integrity Engineering, Chatsworth, CA.

Acknowledgments

The authors wish to acknowledge the guidance and encouragement provided by D. L. Winterhalter of the Office of Space Flight, NASA Headquarters, whose support made this work possible. The authors appreciate the encouragement of R. Bardos, J. Mulcahy, and other individuals of OSF. The review and comments on this work provided by R. Weinstock of Vitro, Washington, D.C., contributed substantially to its successful completion.

The application examples of this report were performed in collaboration with Rocketdyne, Canoga Park, California; Aerojet TechSystems, Nimbus, California; and Pratt & Whitney, West Palm Beach, Florida. Several individuals at each organization contributed generously to this work, including E. P. Fox, C. G. Annis, and D. Paulus of Pratt & Whitney; K. J. O'Hara, D. O'Connor, K. J. Chang, and D. Russell of Rocketdyne; and B. Boehm of Aerojet. The authors worked particularly closely with E. P. Fox of Pratt & Whitney and K. J. O'Hara of Rocketdyne; their considerable contributions are gratefully acknowledged. The contributions of C. G. Annis, D. Russell, and K. J. Chang to the crack growth analysis are also gratefully acknowledged.

Present or former personnel of the NASA Marshall Space Flight Center, whose cooperation was instrumental in successfully performing this work, include J. Lombardo, J. S. Richards, G. W. Smith, L. D. Salter, H. P. Stinson, and J. Townsend.

Throughout the course of this work constructive guidance was provided by the Liquid Rocket Engine Certification Subcommittee of Aerospace Division Committee G-11, Society of Automotive Engineers. The membership of this subcommittee included: W. E. Campbell, Aerojet; K. J. O'Hara, Rocketdyne; E. P. Fox, Pratt & Whitney; J. S. Richards and H. P. Stinson, NASA-MSFC; R. L. Doebler, Aerospace Corp.; and N. R. Moore, JPL.

Finally, the authors wish to acknowledge the review of the technical approach of this work provided by the late R. P. Feynman of the California Institute of Technology.

The authors express their gratitude to all those individuals who contributed to this work and regret that a complete listing is not feasible.

Abstract

An improved methodology for quantitatively evaluating failure risk of spaceflight systems to assess flight readiness and identify risk control measures is presented. This methodology, called Probabilistic Failure Assessment (PFA), combines operating experience from tests and flights with analytical modeling of failure phenomena to estimate failure risk. The PFA methodology is of particular value when information on which to base an assessment of failure risk, including test experience and knowledge of parameters used in analytical modeling, is expensive or difficult to acquire.

The PFA methodology is a prescribed statistical structure in which analytical models that characterize failure phenomena are used conjointly with uncertainties about analysis parameters and/or modeling accuracy to estimate failure probability distributions for specific failure modes. These distributions can then be modified, by means of statistical procedures of the PFA methodology, to reflect any test or flight experience. State-of-the-art analytical models currently employed for design, failure prediction, or performance analysis are used in this methodology.

The PFA methodology can be applied at any time in the design, development, or operational phases of a program to quantitatively estimate failure risk based on the information available at the time failure risk is assessed. Sensitivity analyses conducted as a part of PFA can be used to evaluate and rank such alternative measures to control risk as design changes, testing, or inspections, thereby enabling limited program resources to be allocated more effectively.

PFA is generally applicable to failure modes that can be characterized by analytical or empirical models of failure phenomena and is especially valuable when models or information used in analysis are uncertain or approximate. Such failure modes include, but are not limited to, fatigue, flaw propagation, erosion, malfunctions of mechanical or electrical systems, and shortfalls with respect to performance or life goals for thermal control, electro-optical, power, or propulsion systems.

It is often not feasible to acquire enough test experience to establish high reliability at high confidence for spaceflight systems. Moreover, the results of conventionally performed analytical modeling of failure modes can be subject to serious misinterpretation when uncertain or approximate information is used to establish analysis parameters and calibrate the accuracy of analysis models. Under these conditions, a quantitative evaluation of failure risk based on the information

available from both test or flight experience and analytical modeling is needed to make effective risk management decisions.

This report discusses the rationale for the statistical approach taken in the PFA methodology, describes the PFA methodology, and presents examples of its application to structural failure modes. The engineering models and computer software used in fatigue crack growth and fatigue crack initiation applications are thoroughly documented.

Table of Contents

VOLUME I – Methodology and Applications

VOLUME II – Software Documentation

List of Figures

List of Tables

5.0 Analysis Software

Section 5.1
Crack Growth Analysis Software

5.1.1 Introduction

This section presents a description of the computer program PROCRK which implements the crack growth analysis discussed in Section 2. The code PROCRK was used to analyze the HPOTP Heat Exchanger (HEX) Coil and the proposed External Heat Exchanger (EXHEX). The program PROCRK is modular and hence can be easily modified for crack growth analysis of different components. Different modules were provided for stress analysis and stress intensity factor calculations for the HEX coil and EXHEX analyses. The overall layout of the program is described by using a main flowchart that refers to other flowcharts which describe subprograms and key portions of the main program in greater detail. The program tree structure, a list of subprograms, a description of the key variables, and the FORTRAN source listing for the crack growth analysis code PROCRK are given in Section 7.1. The relevant user's guide for running this code is given in Section 6.1. A glossary of standard flowchart symbols is given in Appendix 5.A.

5.1.2 PROCRK Program

The crack growth methodology is implemented as the FORTRAN program PROCRK. This section provides the description and flowcharts for program PROCRK.

5.1.2.1 Main Routine

The master flowchart for the PROCRK program is given in Figure 5.1-1. The program starts by opening the following input and output files:

NAME	TYPE	CONTENTS
CRKDAT	Input	Analysis data
CRKRES	Output	Input data echo, results
IOUTPR	Output	Run information and user-requested information
LOWLIF	Output	First one percent of sorted crack growth lives

The arrays and variables are set to their default or initial values in the SETDEF routine described in Section 5.1.2.2. The input data is then read from the CRKDAT file in the INPUT routine described in Section 5.1.2.3 and an echo of the input data is written onto the CRKRES file. The materials data including the *da/dN* vs. ΔK crack growth data is read and processed in the GRODAT routine described below in Section 5.1.2.4.

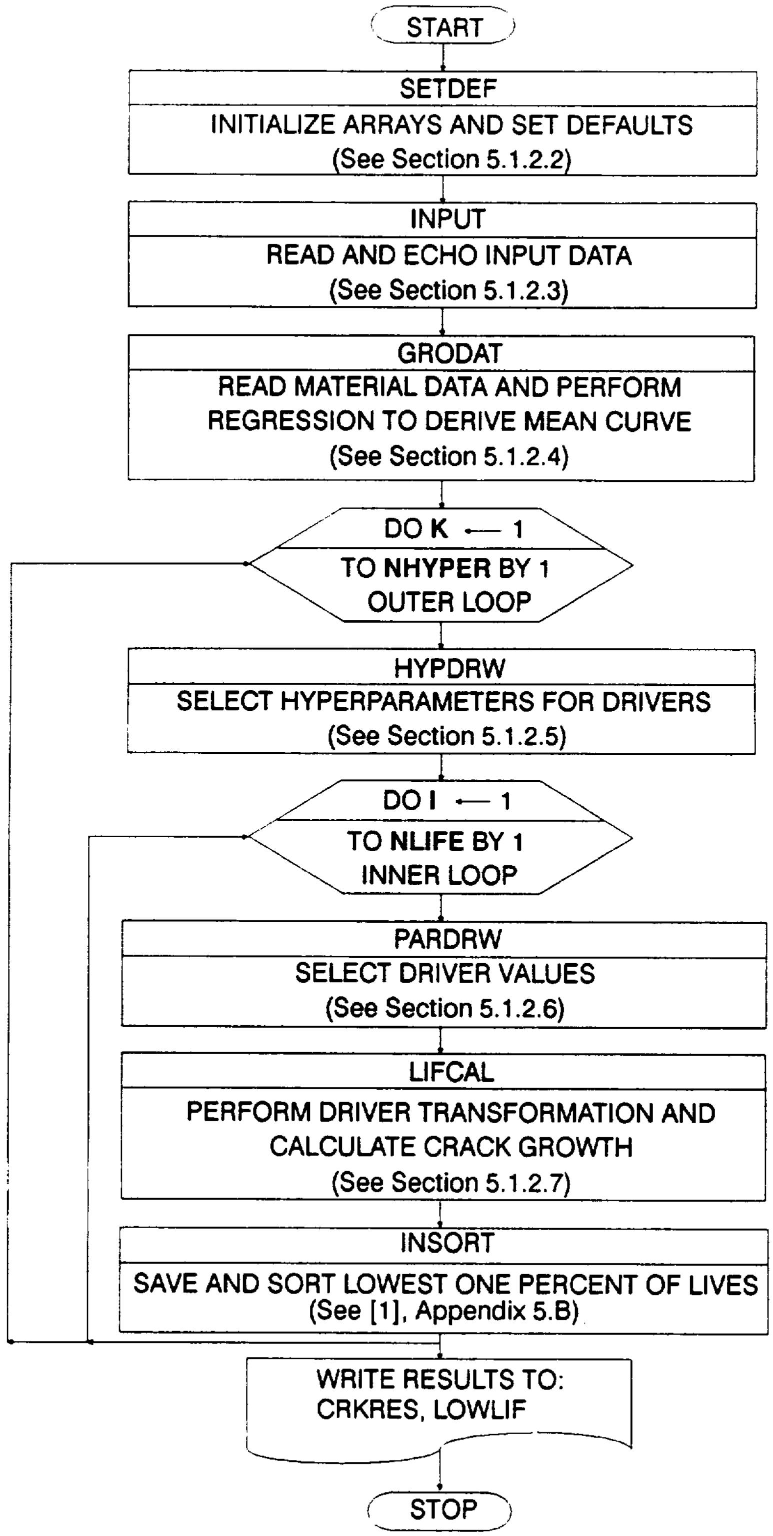


Figure 5.1-1 Main Flowchart for Crack Growth Analysis Program PROCRK

The selection of hyperparameters[1] is performed in the outer DO loop of the simulation by calling the HYPDRW routine described in Section 5.1.2.5. The driver draws are performed within the inner DO loop of the simulation by calling the PARDRW routine described in Section 5.1.2.6. The routine LIFCAL performs driver transformation and calculates the crack growth life within the inner DO loop. The LIFCAL routine is described below in Section 5.1.2.7.

The crack growth lives are arranged in ascending order in a list containing the lowest one percent of the lives. The INSORT routine performs an insertion sort with each new life. When the outer DO loop is completed, the list of lives representing the left-hand tail of the failure distribution is written to file LOWLIF. Routine INSORT is described in Appendix 5.B of [1].

5.1.2.2 SETDEF Routine

The arrays and variables are set to their default or initial values in this routine. Most of the arrays and variables are initialized to zero. The array **LIFE( )**, which is used to store and sort the lowest one percent of the crack growth lives, is initialized to a large value and the number of crack lengths **NCRL** used for block growth calculations is initialized to fifty. Also, the logical variable **FAIL** which flags unstable crack growth (i.e., when $K > K_c$) is initialized to 'FALSE'.

5.1.2.3 INPUT Routine

The input data is read from the CRKDAT file in this routine. First the analysis control variables including the simulation size are read and echoed in the IOUTPR file. Then, the driver distribution information is read. Next, the load/stress history information is read. Finally, some miscellaneous information, such as the Willenborg retardation model parameter, is read. An echo of the input data is written onto the CRKRES file.

5.1.2.4 GRODAT Routine

The flowchart for the GRODAT routine is given in Figure 5.1-2. First the *da/dN* vs. ΔK crack growth data and material properties, such as fracture toughness, threshold stress intensity factor (SIF) range, and tensile strength, are read from the CRKDAT file. Then regression of the crack growth data is performed to fit the generalized Forman Equation 2-7. Four options are available to derive the equation parameters C, n, m, p, and q, as follows:

---

1 Hyperparameters are discussed in Section 2.1.1 of [1].

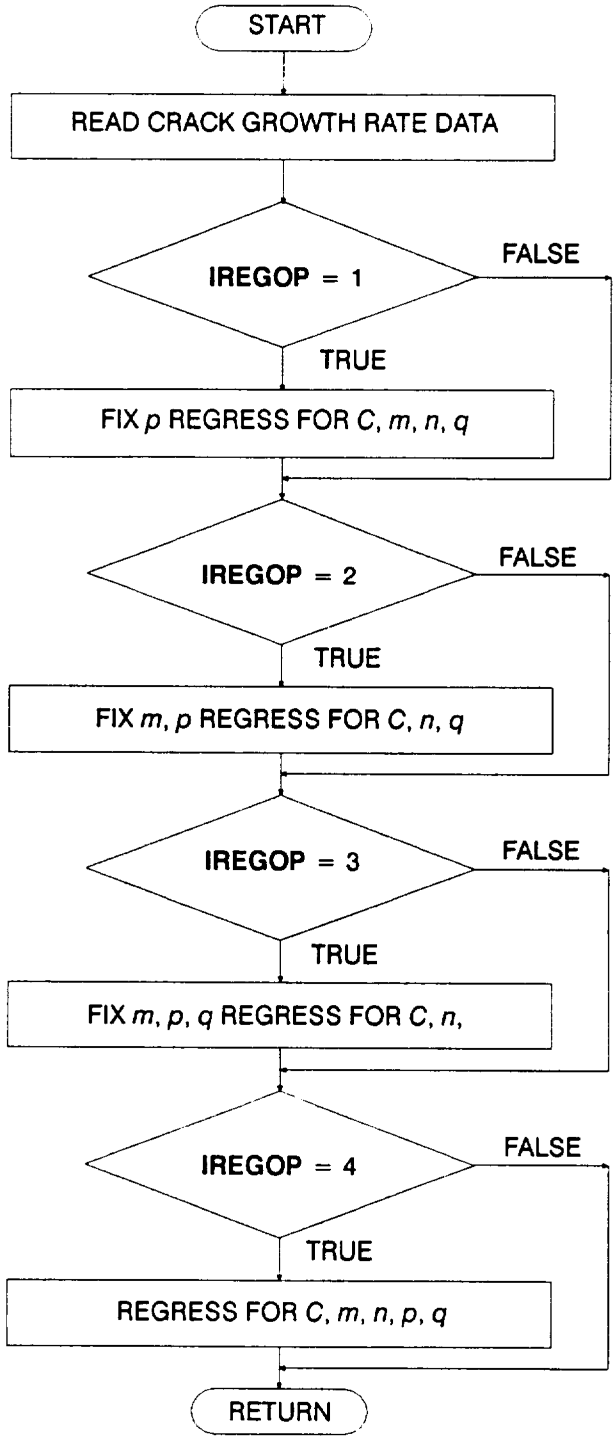


Figure 5.1-2 Flowchart for Subprogram GRODAT

OPTION	PURPOSE
1	Fix p regress for C, m, n, q
2	Fix m, p regress for C, n, q
3	Fix m, p, q regress for C, n
4	Regress for C, m, n, p, q

An external function DETER4 is employed to calculate the determinant of a 4x4 matrix for the **IREGOP** = 4 case.

5.1.2.5 HYPDRW Routine

The selection of hyperparameters is performed in the outer DO loop of the simulation by calling the HYPDRW routine. This includes calling the RANDOM and PRYRV subroutines to obtain the ρ and θ parameters for drivers with Beta distributions, and μ and σ parameters for drivers with Normal distributions.

5.1.2.6 PARDRW Routine

The driver draws are performed within the inner DO loop of the simulation by calling the PARDRW routine. Drivers are selected by calling BETAGN, NORMGN, and PRYRV, which draw from Beta, Normal, and Uniform distributions, respectively. The general-purpose probability distribution subroutines RANDOM, BETAGN, NORMGN, and PRYRV are described in Sections 4.4 and 7.6 of [1].

5.1.2.7 LIFCAL Routine

The flowchart for the LIFCAL routine is given in Figure 5.1-3. First, the stress history is derived in one of the following routines.

ROUTINE	PURPOSE
STRAN1	HEX coil stress calculation
STRAN2	EXHEX stress calculation

STRAN1 and STRAN2 routines are described in Section 5.1.2.8. A rainflow cycle count is performed and a stress-level vs. number-of-cycles table is generated in the CYCOUN routine described in Section 5.1.2.9. The life integration DO loop calculates block growth rates at **NCRL** number of crack lengths by calling the BLKGRO routine described in Section 5.1.2.10.

5.1.2.8 STRAN1 and STRAN2 Routines

The flowchart for the STRAN1 routine is given in Figure 5.1-4. The maximum principal stress was assumed to be the axial stress component for the HEX coil. The composite principal stress history, which is due to static, random, sinusoidal, and aerodynamic loads, is derived in this routine. First, the static stresses are assigned to the stress history. Then, the reference time histories for each load component are scaled by the non-time-varying dynamic stress magnitudes and added

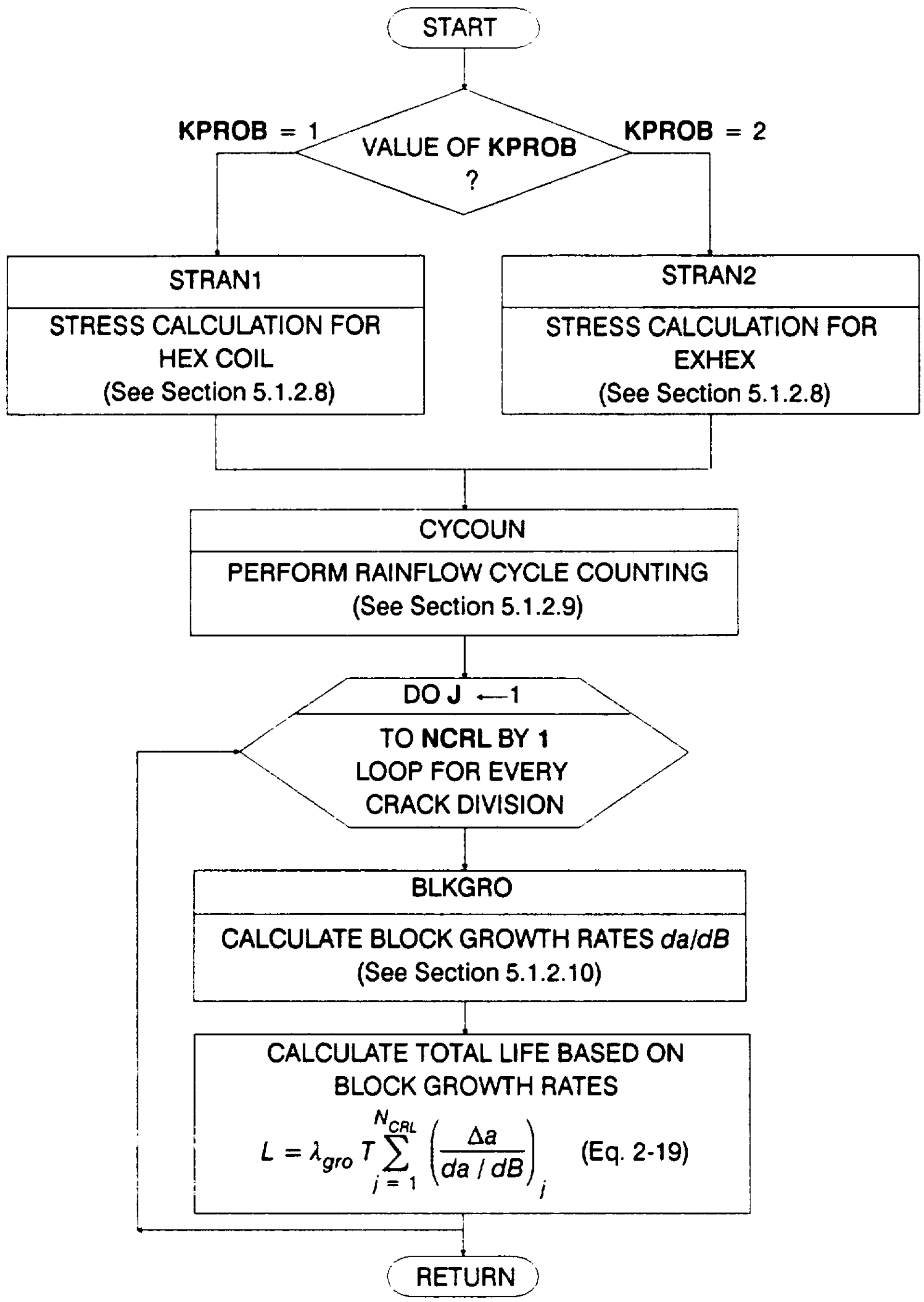


Figure 5.1-3 Flowchart for Subprogram LIFCAL

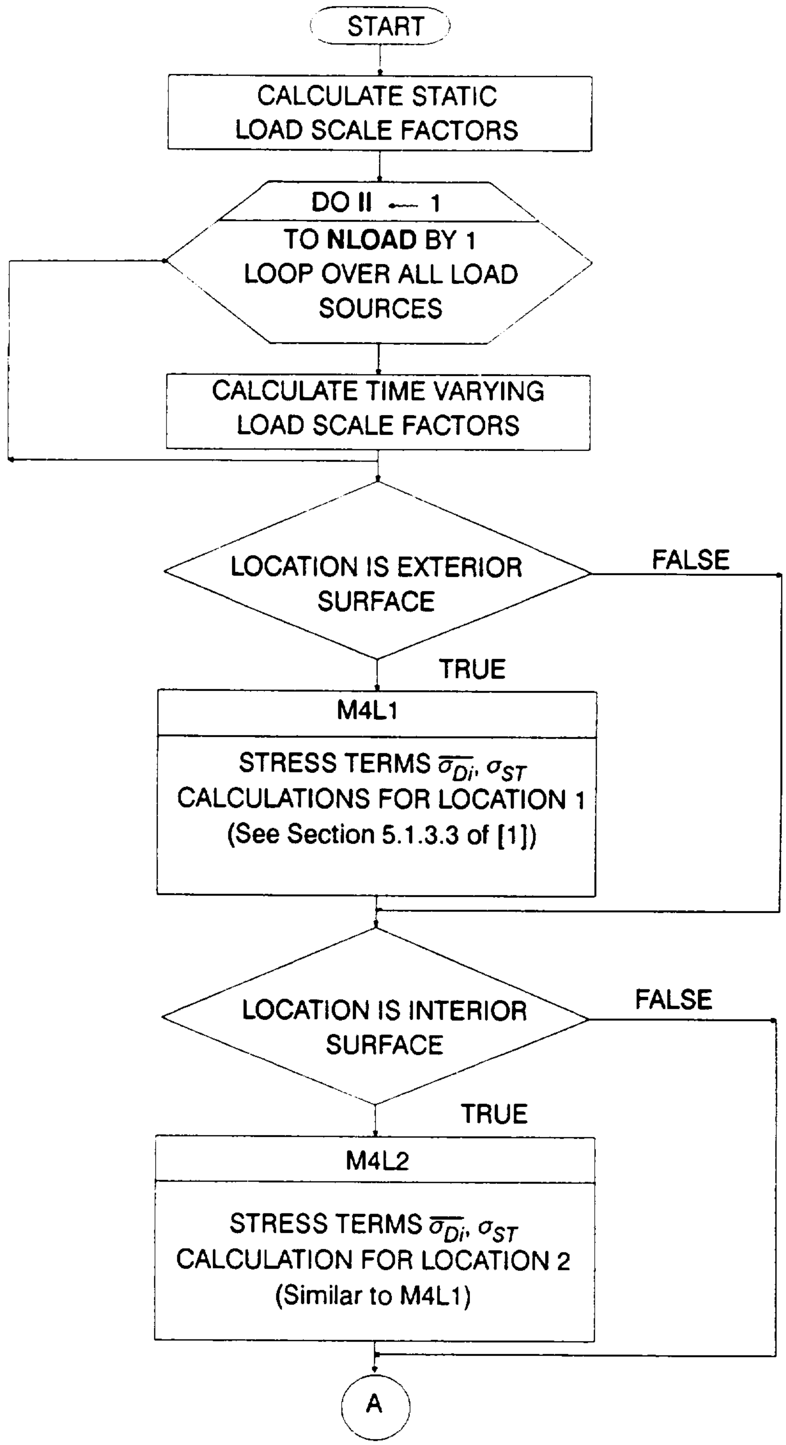


Figure 5.1-4 Flowchart for Subprogram STRAN1

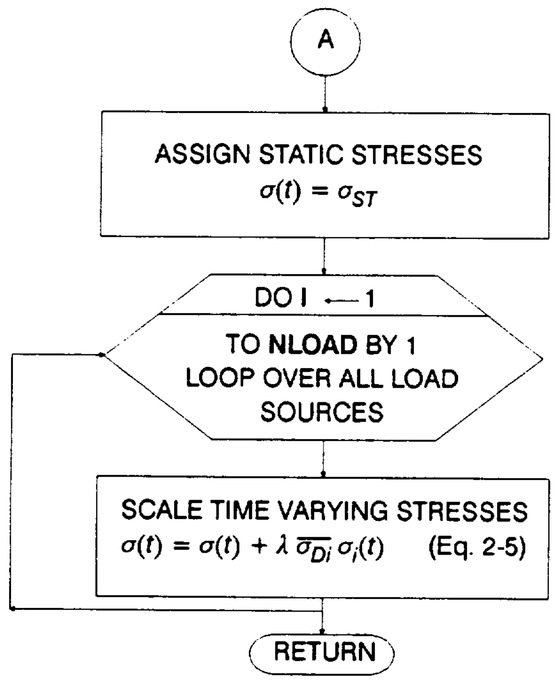


Figure 5.1-4 Flowchart for Subprogram STRAN1 (Cont'd)

to the principal stress time history. The stress magnitudes are calculated by calling the following routines.

LOCATION	POSITION	ROUTINE
1	Exterior Surface	M4L1
2	Interior Surface	M4L2

M4L1 and M4L2 routines are described in Section 5.1.2.3 in [1].

The flowchart for the STRAN2 routine is given in Figure 5.1-5. This routine is similar to STRAN1 except that the stress magnitudes rather than the load magnitudes are provided as input and hence additional routines are not called for the stress magnitude calculations. Due to the nature of the loading in the EXHEX the maximum principal stress was assumed to be equal to the σ_z component.

5.1.2.9 CYCOUN Routine

The flowchart for CYCOUN is given in Figure 5.1-6. This routine is similar to the rainflow cycle counting routine described in Section 5.1.3.5 in [1].

First, the principal stress history is scanned to identify the largest stress and its location. The history is resequenced such that the largest stress is placed at the beginning and end of the stress array. Then, the intermediate points in the history are filtered, leaving only the peaks and troughs. This is done by testing for a sign change in the gradients of adjacent segments. Next, the counting of the cycles begins. Consecutive peaks and troughs are added to a holding array, each time checking whether the new peak-trough segment is greater than the previous one; if so, then a cycle has been closed. Then, the peak and trough corresponding to the closed cycle are removed from the holding array. The cycle is saved if it is large enough, i.e., larger than a user-specified threshold. The procedure is repeated by adding new peaks and troughs to the holding array until another cycle is identified.

Once all the cycles have been identified, the alternating and mean values of each stress cycle are calculated. The stress range of the biggest cycle is divided into one hundred equal stress ranges (or bins) and each stress cycle is assigned to a bin based on its magnitude. This reduces the results of the cycle counting to a number-of-cycles vs. stress-level table. An equivalent mean stress may be calculated for the entire history based on the mean of the biggest cycle. The routine NEUBER, described in Section 5.1.3.6 in [1], is called to estimate the equivalent mean stress.

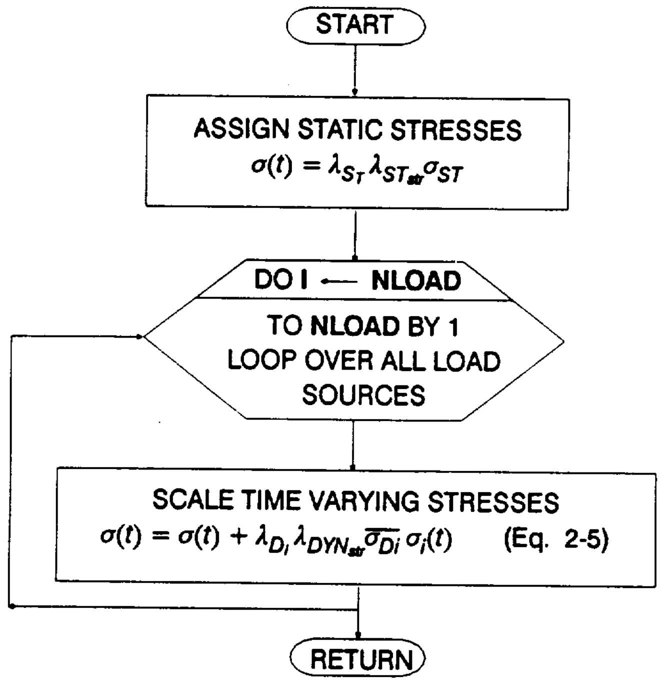


Figure 5.1-5 Flowchart for Subprogram STRAN2

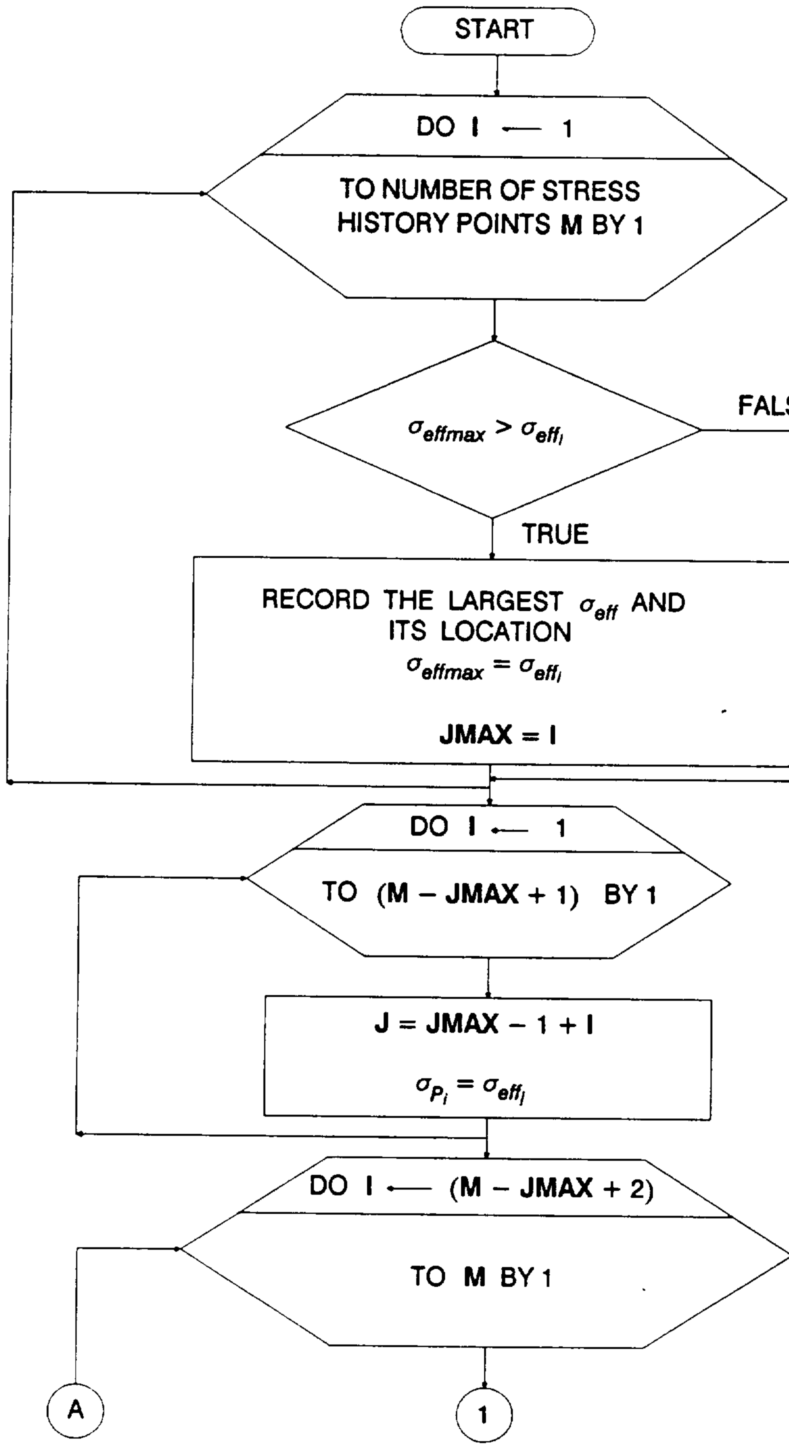


Figure 5.1-6 Flowchart for Subprogram CYCOUN

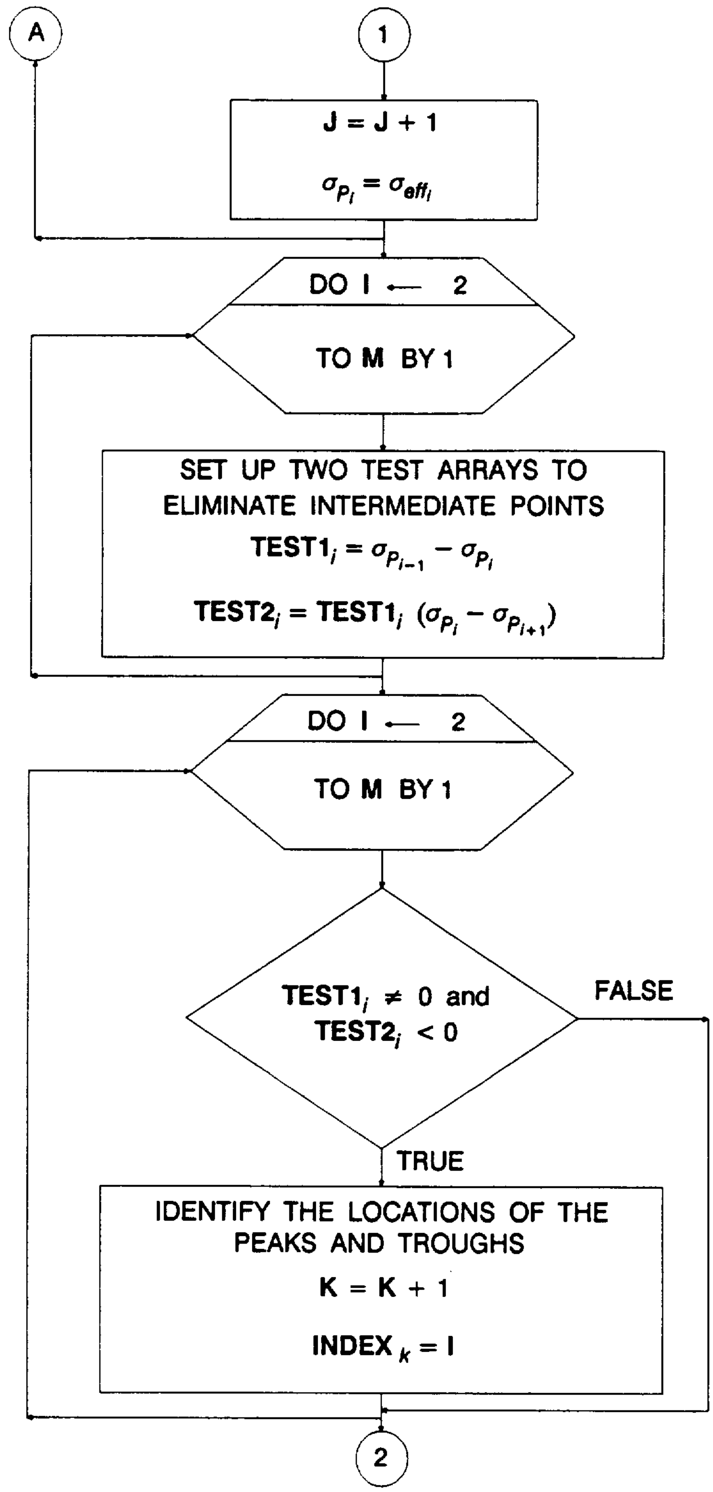


Figure 5.1-6 Flowchart for Subprogram CYCOUN (Cont'd)

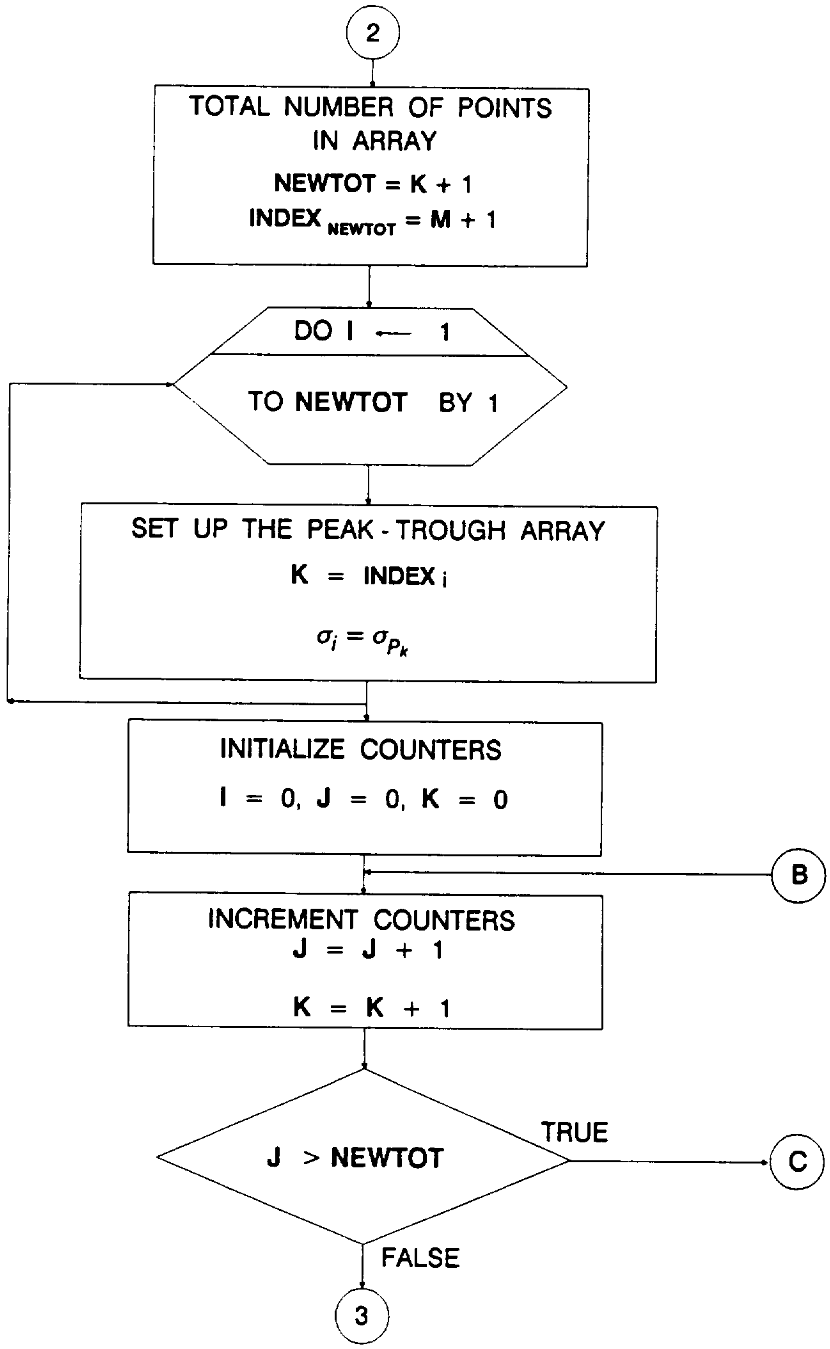


Figure 5.1-6 Flowchart for Subprogram CYCOUN (Cont'd)

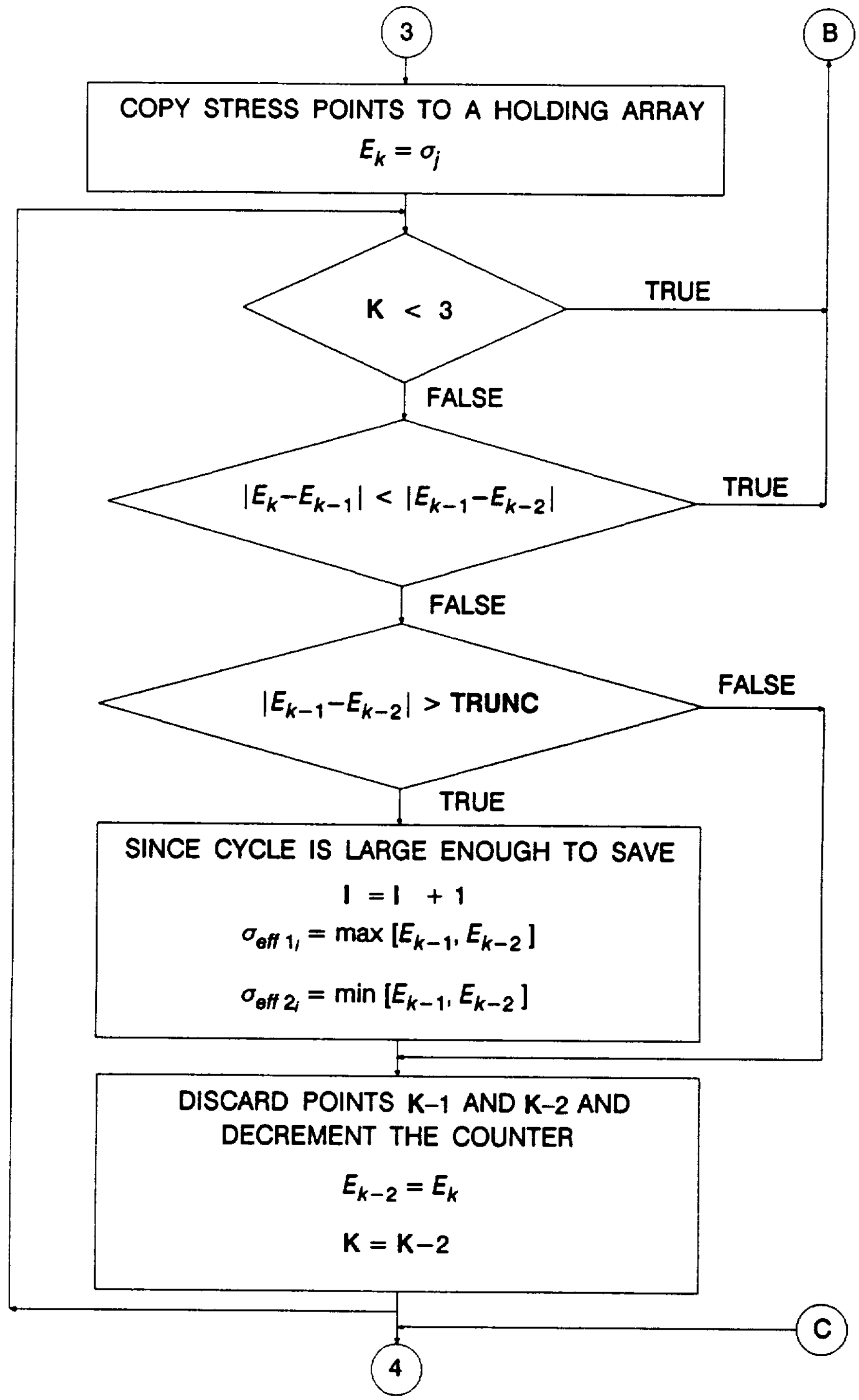


Figure 5.1-6 Flowchart for Subprogram CYCOUN (Cont'd)

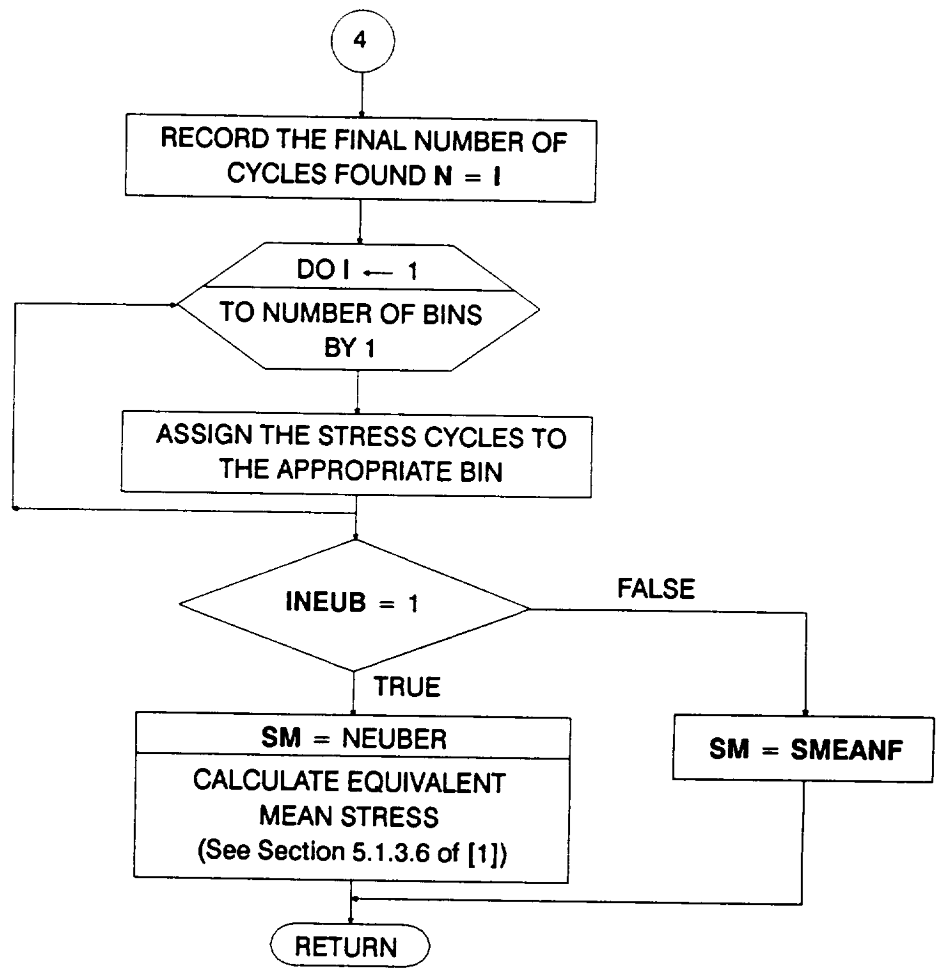


Figure 5.1-6 Flowchart for Subprogram CYCOUN (Cont'd)

5.1.2.10 BLKGRO Routine

The flowchart for BLKGRO is given in Figure 5.1-7. First, the stress-intensity factor coefficients are calculated in the following routines:

ROUTINE	PURPOSE
STRIF1	Stress intensity factor coefficients for HEX coil
STRIF2	Stress intensity factor coefficients for EXHEX

The stress intensity factor routines STRIF1 and STRIF2 are described in Section 5.1.2.11.

The crack growth in a block is calculated as given by Equation 2-17, by summing the growth due to the cycles at each stress level, for each direction (*a* and *c*) of crack growth. If growth retardation is considered (**IRET** = 1), an effective SIF range, ΔK_{eff}, and stress ratio, R_{eff}, are calculated as per the Willenborg model given by Equations 2-12 through 2-16. Growth calculations are performed after checking for $\Delta K_{eff} < \Delta K_{th}$ and $K_{max} > K_c$ conditions, which are the no-growth and the unstable crack cases, respectively.

5.1.2.11 STRIF1 and STRIF2 Routines

The STRIF1 routine calculates stress intensity factor coefficients for the HEX coil crack configuration. As described in Section 2.1 the standard solution, for an elliptic crack in a finite width plate, given in NASA/FLAGRO [2] is employed.

The STRIF2 routine calculates SIF coefficients for the EXHEX crack configuration. The expressions given in [3] for a crack in a plate are employed.

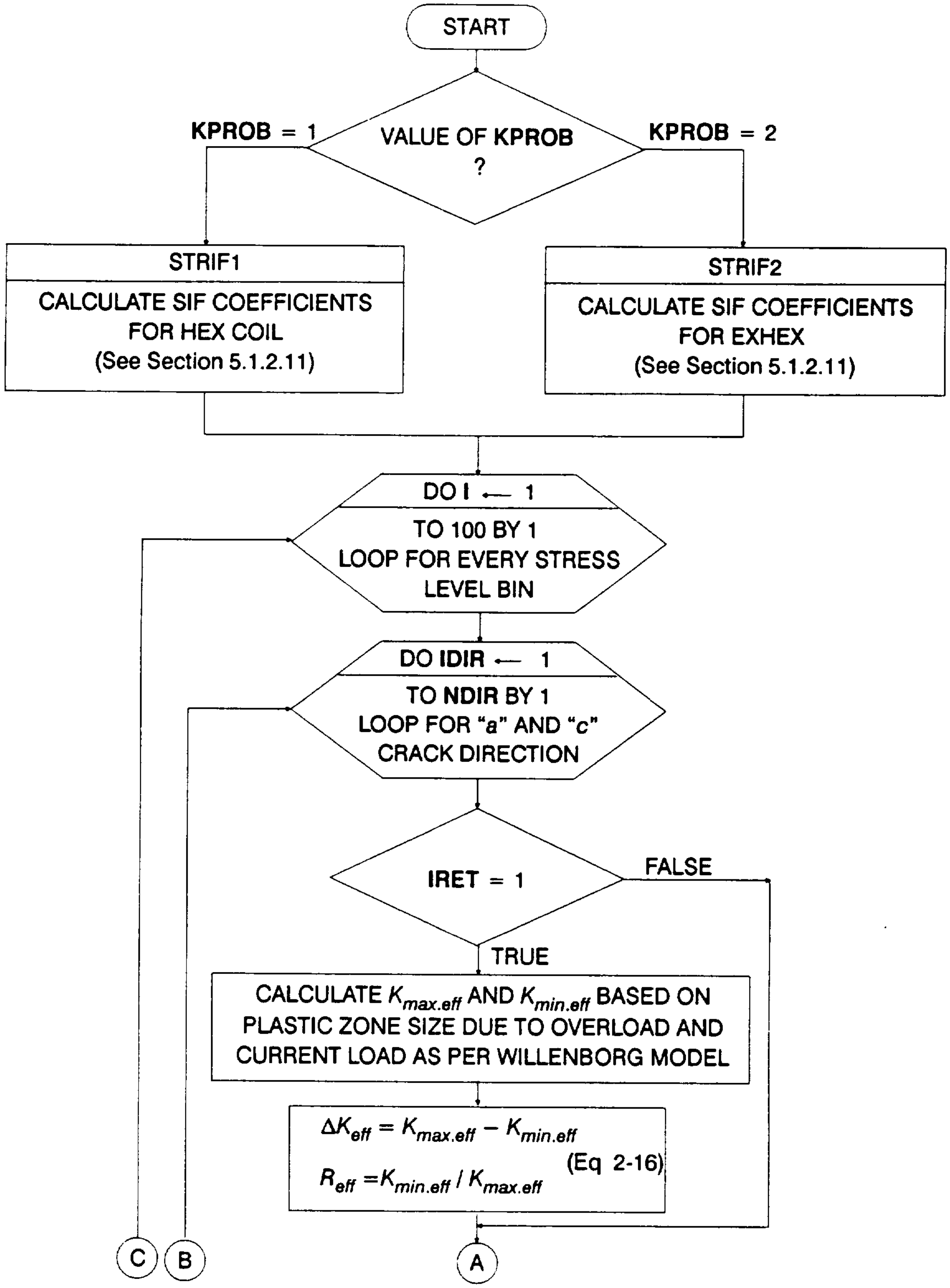


Figure 5.1-7 Flowchart for Subprogram BLKGRO

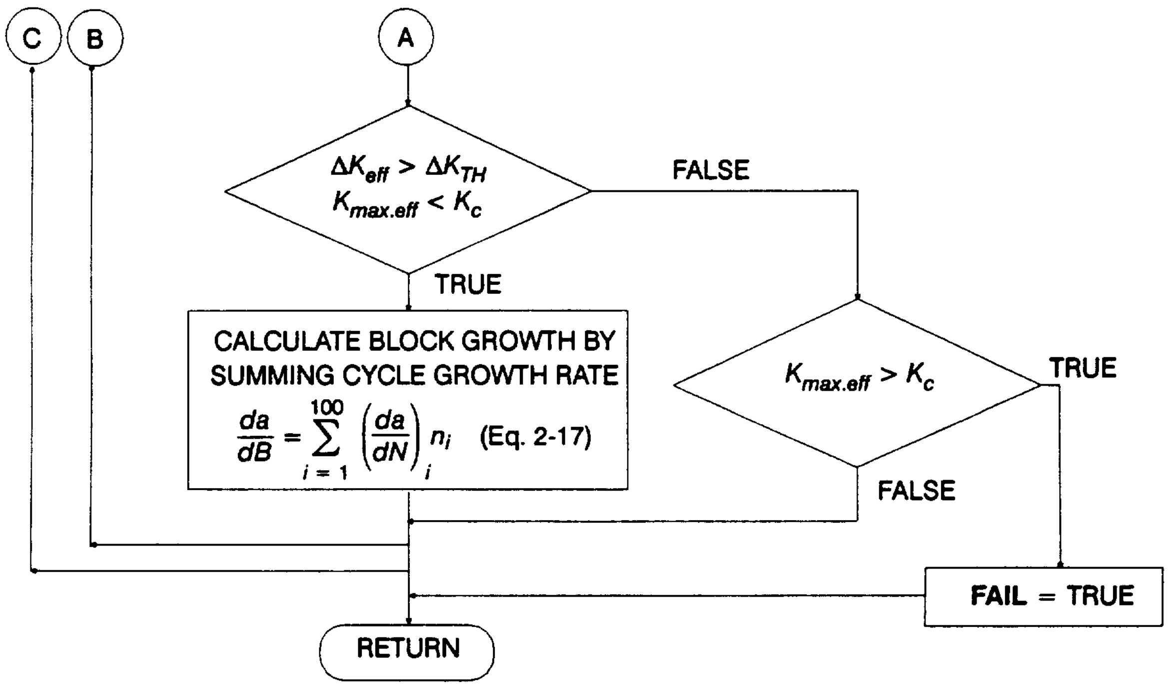


Figure 5.1-7 Flowchart for Subprogram BLKGRO (Cont'd)

Section 5.2
Low Cycle Fatigue Analysis Software

5.2.1 Introduction

This section presents a description of the computer program which implements the LCF analysis discussed in Section 3.2. The code for analyzing the ATD-HPFTP first stage turbine blade is described below in Section 5.2.2. The overall layout of the program is described by using a main flowchart that refers to other flowcharts, which describe subprograms and key portions of the main program in greater detail. The program tree structure, a list of subprograms, a description of the key variables, and the FORTRAN source listing for the LCF analysis code BLDLCF are given in Section 7.2. The materials characterization subprograms and those subprograms that are of a generic nature, such as the random variate generators, are described in [1], Section 4.1 and Section 4.4 respectively. The relevant user's guide for running this code is given in Section 6.2. A glossary of standard flowchart symbols is given for the reader's benefit in Appendix 5.A.

5.2.2 BLDLCF Program

The LCF analysis of the ATD-HPFTP first stage turbine blade is implemented as the FORTRAN program BLDLCF. Figure 5.2-1 shows the structure of the Probabilistic Failure Model (PFM) for the Blade. This section provides the description and flowcharts for program BLDLCF.

5.2.2.1 Main Routine

The master flowchart for the BLDLCF program is given in Figure 5.2-2. The program starts by opening the following input and output files:[2]

NAME	TYPE	CONTENTS
BLDLCD	Input	Analysis data
BLDLCO	Output	Input data echo, results
RELATD	Input	Related material data input
RELATO	Output	Echo of information in RELATD
DUMP	Output	Results of materials characterization calculations
IOUTPR	Output	Run information and user-requested information
LOWLIF	Output	First one percent of sorted fatigue lives

The arrays and variables are then set to their default or initial values. The input data is read from the BLDLCD file. An echo of the input data is written onto BLDLCO. The related material S/N information is read from the file RELATD and

[2] Files RELATD and RELATO are opened in INFAGG.

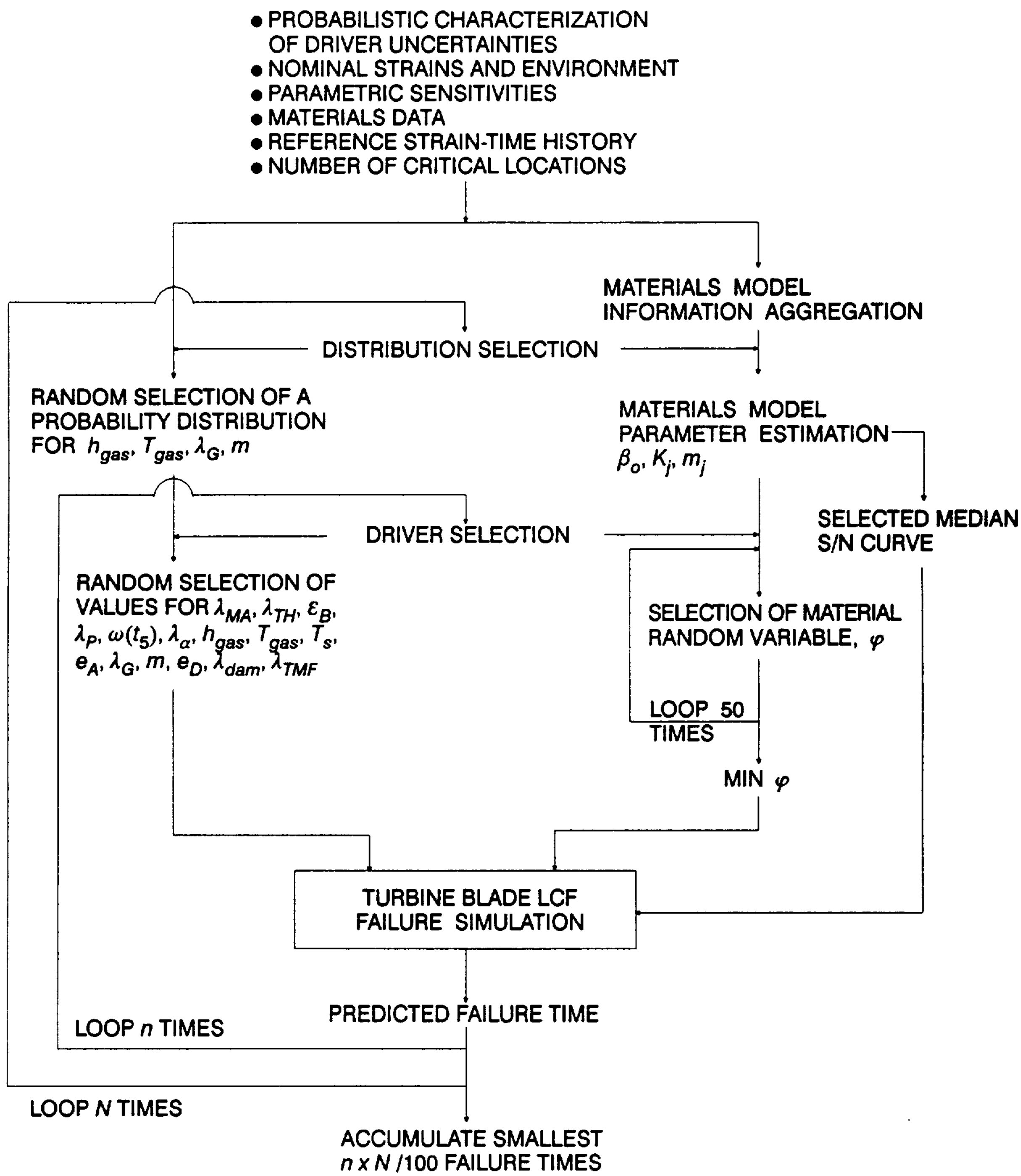


Figure 5.2-1 Structure of the Turbine Blade LCF Probabilistic Failure Model

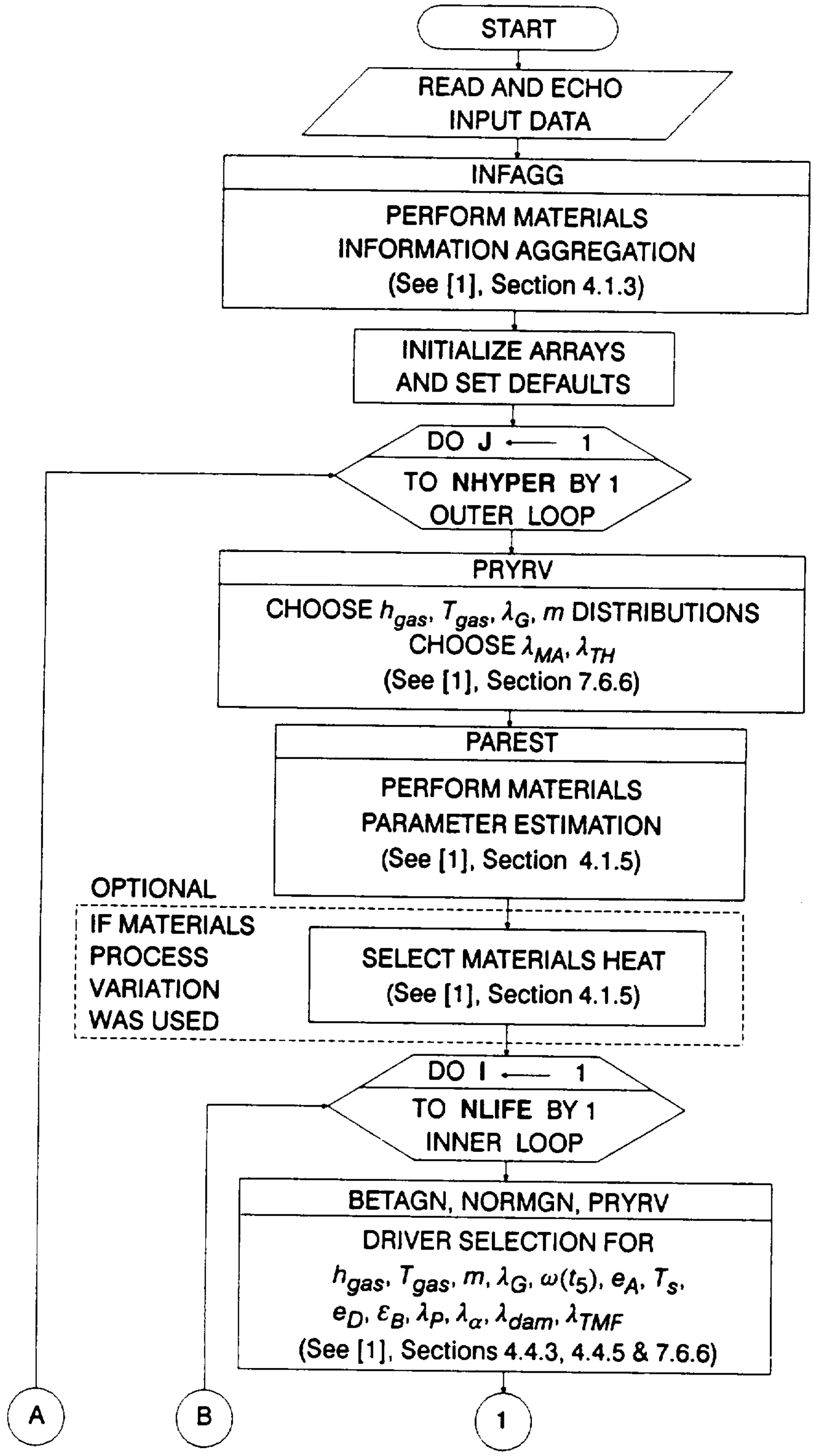


Figure 5.2-2 Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF

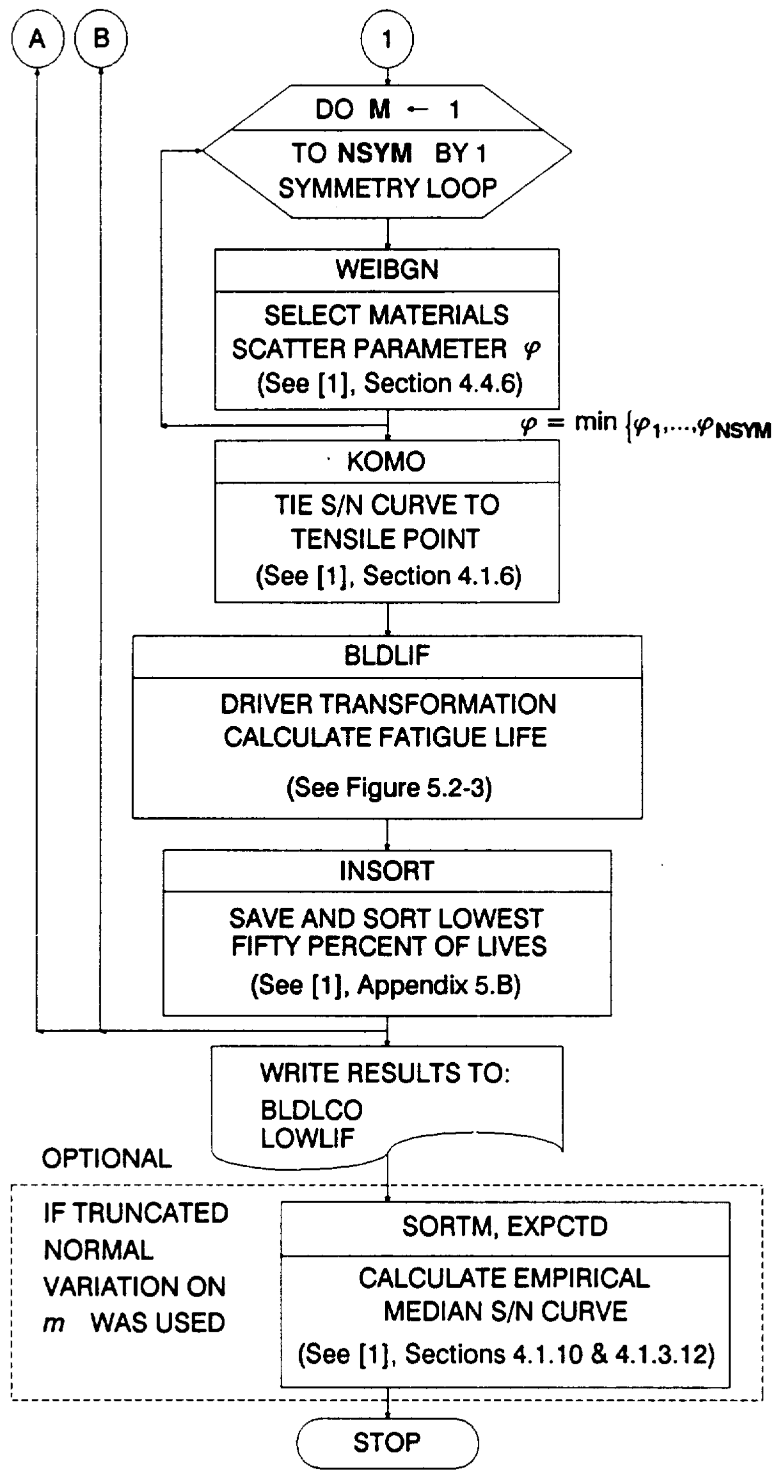


Figure 5.2-2 Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF (Cont'd)

processed in the INFAGG routine. INFAGG controls the materials information aggregation and is described in [1], Section 4.1.3.

The selection of hyperparameters[3] is performed in the outer DO loop of the simulation by calling the PRYRV routine to obtain the Beta distribution parameters ρ and θ for h_{gas}, T_{gas}, m, and λ_G, whose probability distributions are described by Beta distributions. The selection of values for λ_{MA} and λ_{TH} is also performed. The PAREST routine controls the calculations for estimating the parameters for the S/N model. Routine PAREST is described in [1], Section 4.1.5. If materials process variation is included, the materials parameter Z in [1], Equation 2-48 is selected by calling the NORMGN routine and then transforming the resulting Normal variate to a Lognormal variate.

The inner DO loop for the simulation performs the driver selection. The drivers h_{gas}, T_{gas}, m, λ_G, $\omega(t_5)$, e_A, T_s, e_D, ε_B, λ_P, λ_α, λ_{dam}, and λ_{TMF} are selected by calling BETAGN, NORMGN, and PRYRV which draw from Beta, Normal, and Uniform distributions, respectively. The random variate routines BETAGN, NORMGN, and PRYRV are described in [1], Sections 4.4, and 7.6.

In the symmetry DO loop, the materials model parameter φ is found from the minimum of 50 draws of a Weibull distribution. Calls to WEIBGN provide the 50 values of φ. Subroutine WEIBGN is described in [1], Section 4.4.6.

When all the S/N model parameters have been selected for the region with S/N data, the S/N curve is tied to the tensile point S_o by routine KOMO. The routine BLDLIF performs driver transformation and calculates the fatigue life. The flowchart for BLDLIF is given in Figure 5.2-3 and the routine is described below. Subprogram KOMO is described in [1], Sections 4.1.6.

The fatigue lives are arranged in ascending order in a list containing the lowest fifty percent of the lives. The INSORT routine performs an insertion sort with each new fatigue life. When the outer DO loop is completed, the list of lives representing the left-hand tail of the failure distribution is written to file LOWLIF. Subprogram INSORT is described in [1], Appendix 5.B.

If a truncated Normal distribution was used for the materials shape parameter m, the empirical median S/N curve will be calculated upon user request. The routine SORTM is called to sort the values of m and the routine EXPCTD calculates the

[3] Hyperparameters are discussed in [1], Section 2.1.1.

median S/N curve. Sections 4.1.10 and 4.1.3.12 of [1], describe the routines SORTM and EXPCTD, respectively.

5.2.2.2 BLDLIF Routine

The flowchart for the BLDLIF routine is given in Figure 5.2-3. First, the thermal strain during acceleration is calculated using the acceleration model of Equation 3-2. Next, the deceleration model calculations are performed, Equations 3-3, 3-6, and 3-7, the deceleration slope, thermal strain, and rotor speed are obtained. The total mechanical and total thermal strain-time histories are calculated using Equations 3-5 and 3-4, respectively. Then, the composite strain-time history is obtained by combining the thermal and mechanical strains according to Equation 3-1. Finally, the RAINF3 routine is called. This routine performs a rainflow cycle count and derives the fatigue life.

5.2.2.3 RAINF3 Routine

The flowchart for RAINF3 is given in Figure 5.2-4. First, the equivalent strain history is scanned to identify the largest strain and its location. The history is resequenced such that the largest strain is placed at the beginning and end of the strain array. Then, the intermediate points in the history are filtered leaving only the peaks and troughs. This is done by testing for a sign change in the gradients of adjacent segments. Next, the counting of the cycles begins. Consecutive peaks and troughs are added to a holding array, each time checking if the new peak-trough segment is greater than the previous one; if so, then a cycle has been closed. Then, the peak and trough corresponding to the closed cycle are removed from the holding array. The cycle is saved if it is large enough, i.e., larger than a user-specified threshold. The procedure is repeated by adding new peaks and troughs to the holding array until another cycle is identified.

Once all the cycles have been identified, an equivalent zero-mean strain range is calculated for each cycle using the Walker relation given by Equation 3-8. The life corresponding to each strain cycle is obtained from the S/N curve by calling the GTLIFE routine. The GTLIFE routine is described under materials characterization in [1], Section 4.1.8. Miner's rule is used to accumulate the damage due to each cycle. There are three separate DO loops over the number of cycles in the last three steps, starting with the Walker transformation. This was done to enable vectorization of the DO loops. For running on a scalar machine, these three steps may be embedded within a single DO loop.

5.2.3 BLDLCF Program, Nonparametric Materials Model

The LCF analysis of the ATD-HPFTP first stage turbine blade using the nonparametric materials model is implemented as the FORTRAN program BLDLCF V3.4B1.3. Figure 5.2-5 shows the structure of the PFM for the Blade using the non-

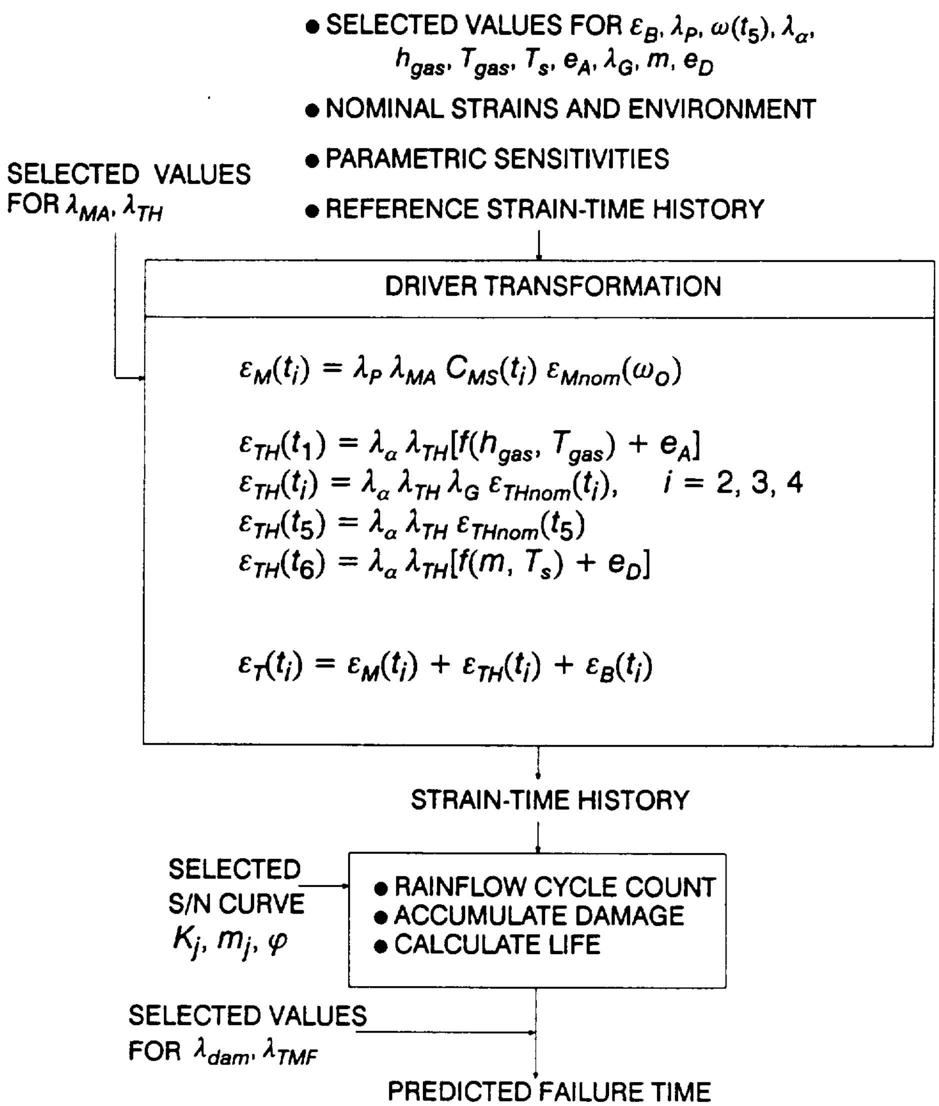


Figure 5.2-3 Flowchart for Subprogram BLDLIF

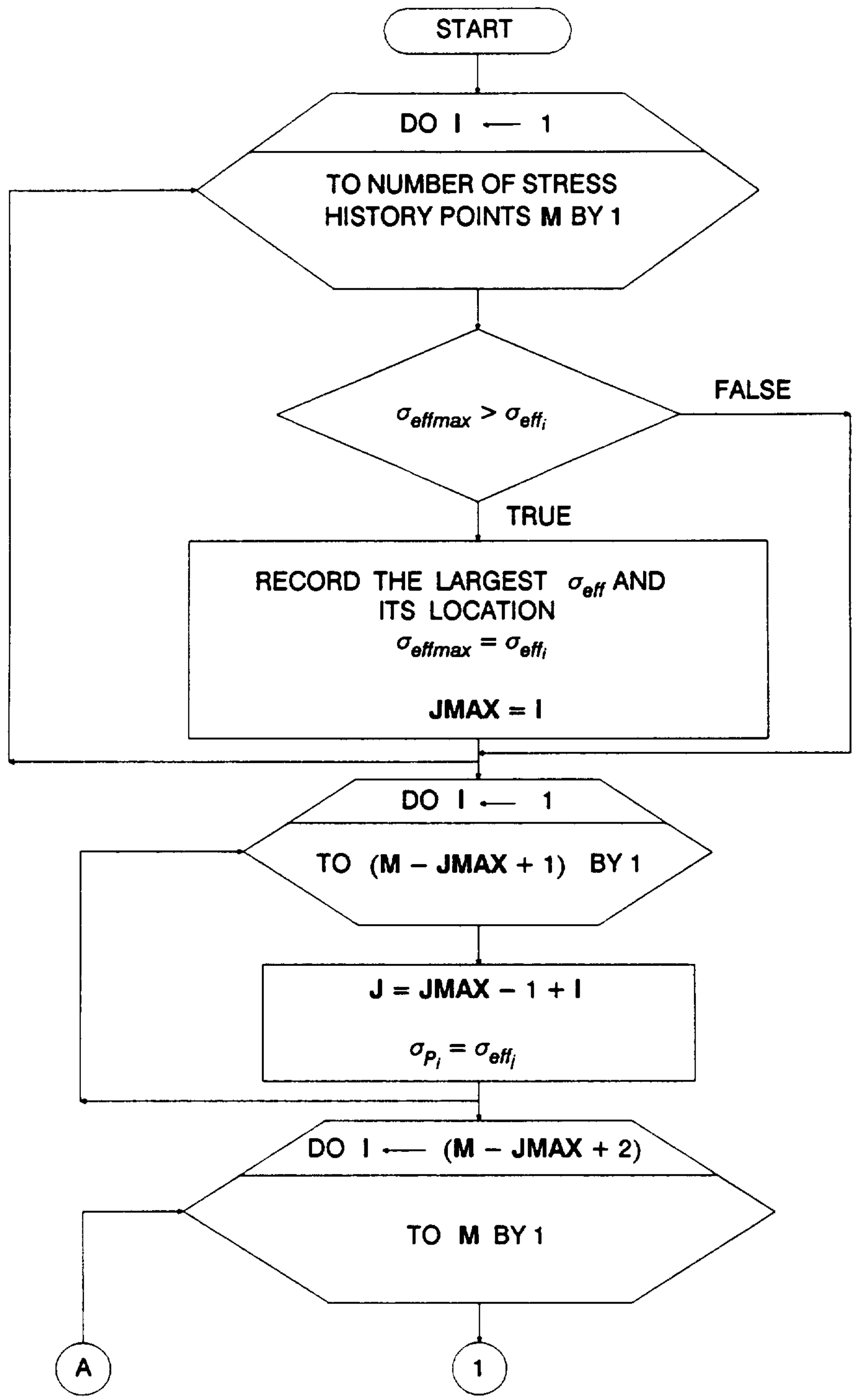


Figure 5.2-4 Flowchart for Subprogram RAINF3

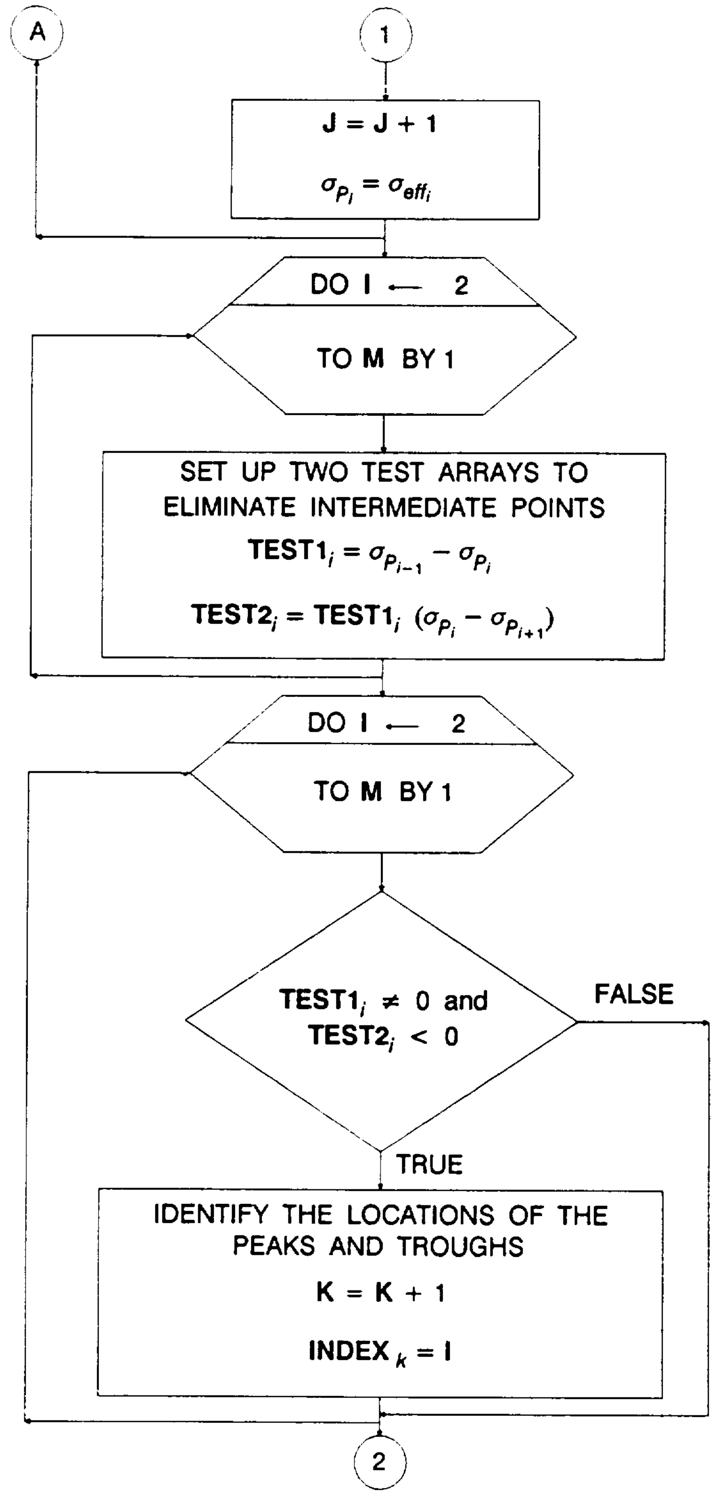


Figure 5.2-4 Flowchart for Subprogram RAINF3 (Cont'd)

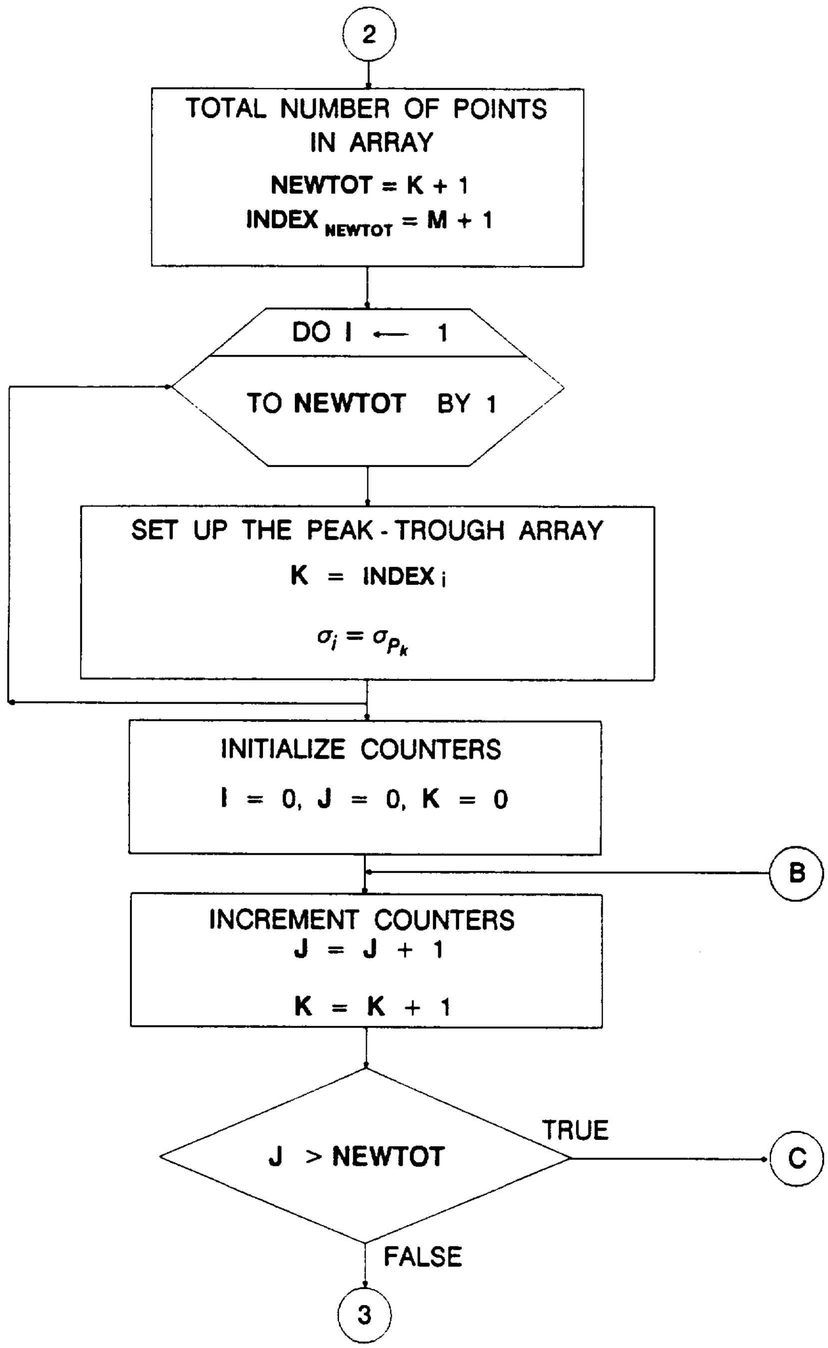


Figure 5.2-4 Flowchart for Subprogram RAINF3 (Cont'd)

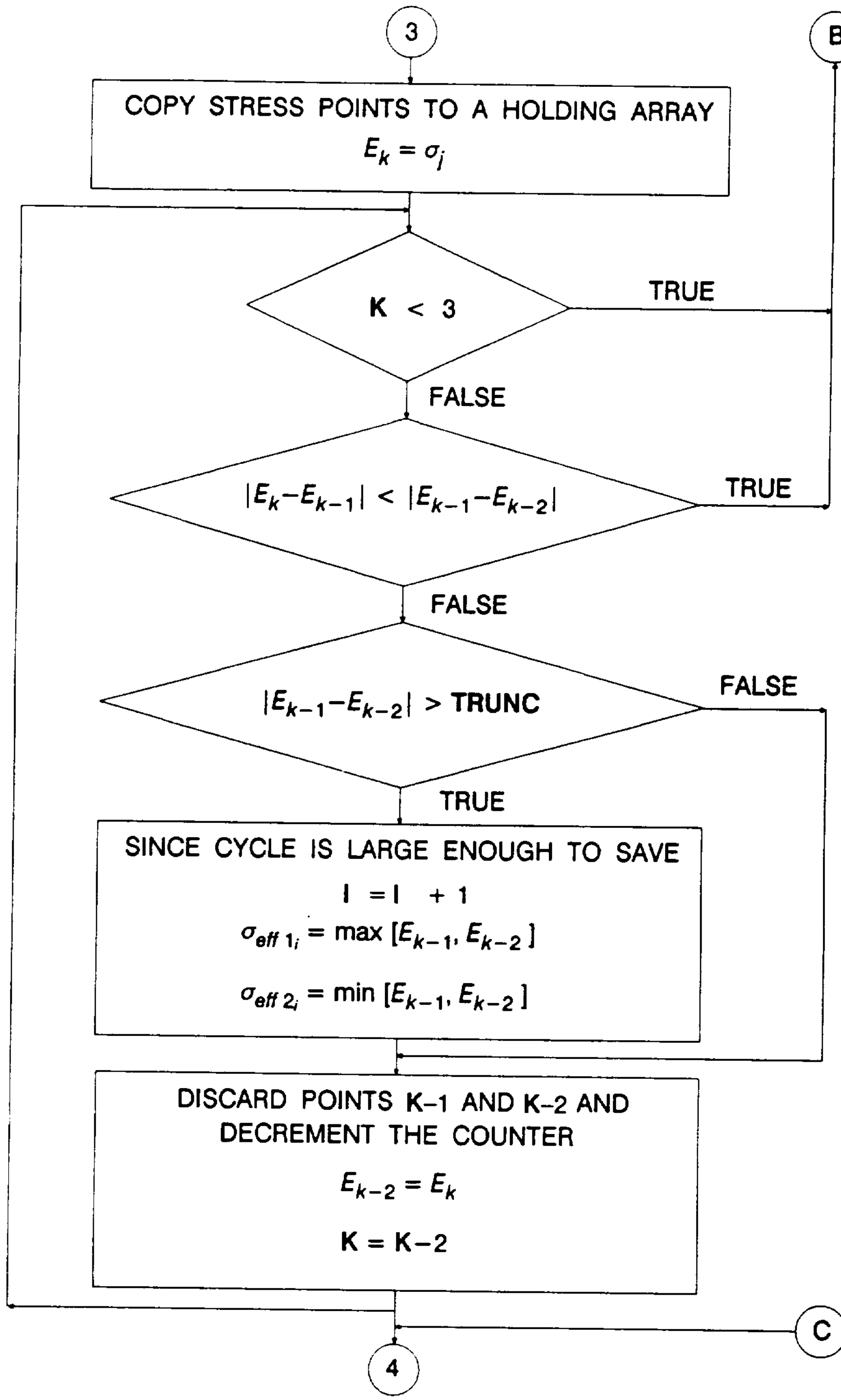


Figure 5.2-4 Flowchart for Subprogram RAINF3 (Cont'd)

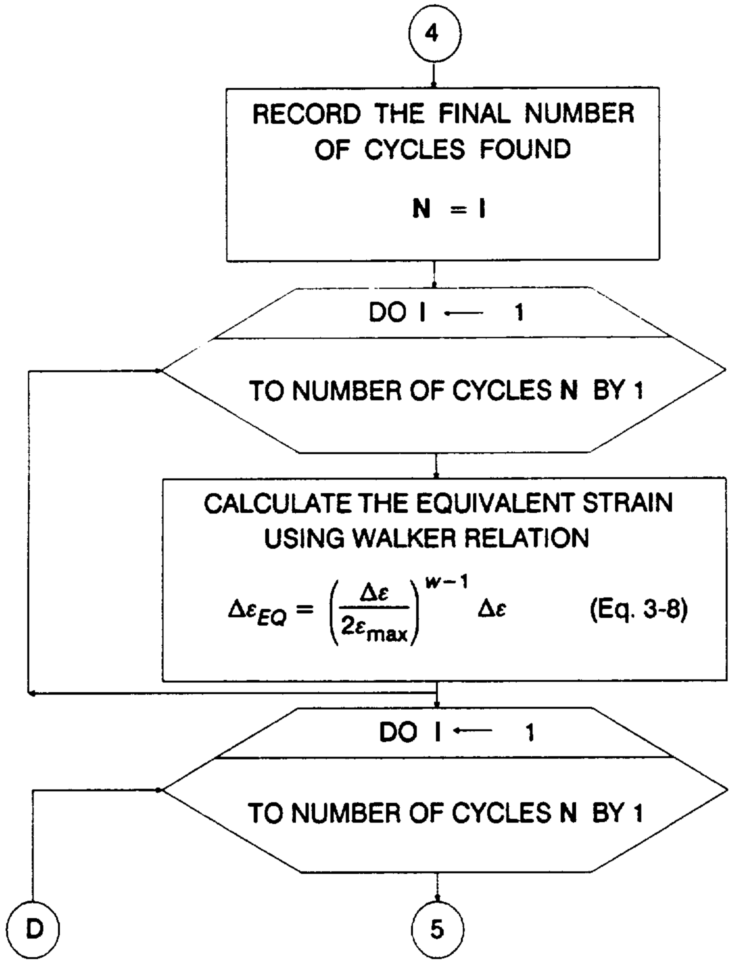


Figure 5.2-4 Flowchart for Subprogram RAINF3 (Cont'd)

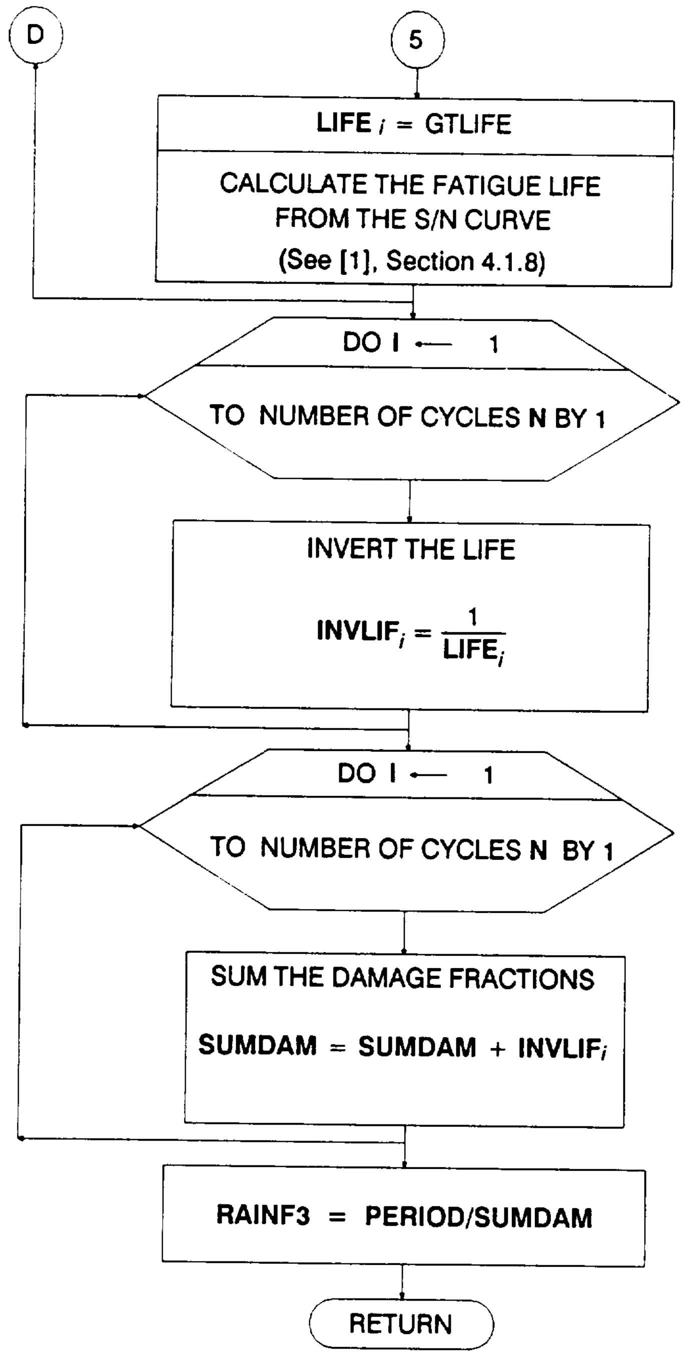


Figure 5.2-4 Flowchart for Subprogram RAINF3 (Cont'd)

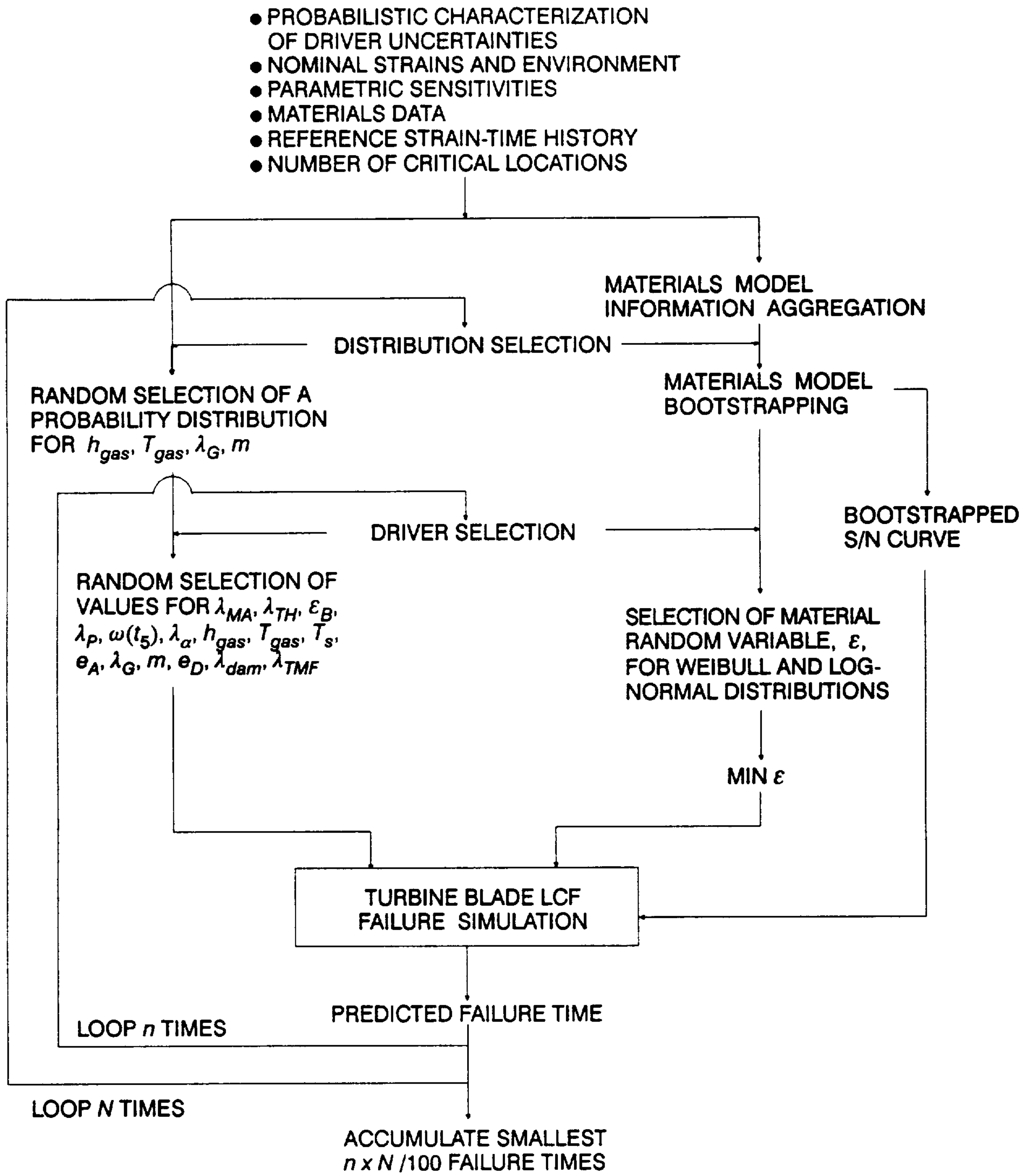


Figure 5.2-5 Structure of the Turbine Blade LCF Probabilistic Failure Model Using the Nonparametric Materials Model

parametric materials model. This section provides the description and flowcharts for program BLDLCF V3.4B1.3 and its routines which differ from Section 5.2.2 above and Section 4.1 of [1].

5.2.3.1 Main Routine

The master flowchart for the BLDLCF V3.4B1.3 program is given in Figure 5.2-6. The program starts by opening the following input and output files:[4]

NAME	TYPE	CONTENTS
BLDLCD	Input	Analysis data
BLDLCO	Output	Input data echo, results
RELATD	Input	Related material data input
RELATO	Output	Echo of information in RELATD
DUMP	Output	Results of materials characterization calculations
IOUTPR	Output	Run information and user-requested information
LOWLIF	Output	First one percent of sorted fatigue lives

The arrays and variables are then set to their default or initial values. The input data is read from the BLDLCD file. An echo of the input data is written onto BLDLCO. The related material S/N information is read from the file RELATD and processed in the INFAGG routine. INFAGG controls the materials information aggregation and is described in Section 5.2.3.2.

The selection of hyperparameters[5] is performed in the outer DO loop of the simulation by calling the PRYRV routine to obtain the Beta distribution parameters ρ and θ for h_{gas}, T_{gas}, m, and λ_G, whose probability distributions are described by Beta distributions. The selection of values for λ_{MA} and λ_{TH} is also performed. The PEB routine controls the calculations for bootstrapping the residuals, generating the pseudo S/N data, and then calculating the structural parameters. Routine PEB is described in Section 5.2.3.5.[6] If materials process variation is included, the materials parameter Z in [1], Equation 2-48 is selected by calling the NORMGN routine and then transforming the resulting Normal variate to a Lognormal variate.

The inner DO loop for the simulation performs the driver selection. The drivers h_{gas}, T_{gas}, m, λ_G, $\omega(t_5)$, e_A, T_s, e_D, ε_B, λ_P, λ_α, λ_{dam}, and λ_{TMF} are selected by calling BETAGN, NORMGN, and PRYRV, which draw from Beta, Normal, and Uniform distributions, respectively. The random variate routines BETAGN, NORMGN, and PRYRV are described in [1], Sections 4.4, and 7.6.

---

4 Files RELATD and RELATO are opened in INFAGG.

5 Hyperparameters are discussed in [1], Section 2.1.1.

6 The bootstrapping calculations are discussed in Section 3.2.7.

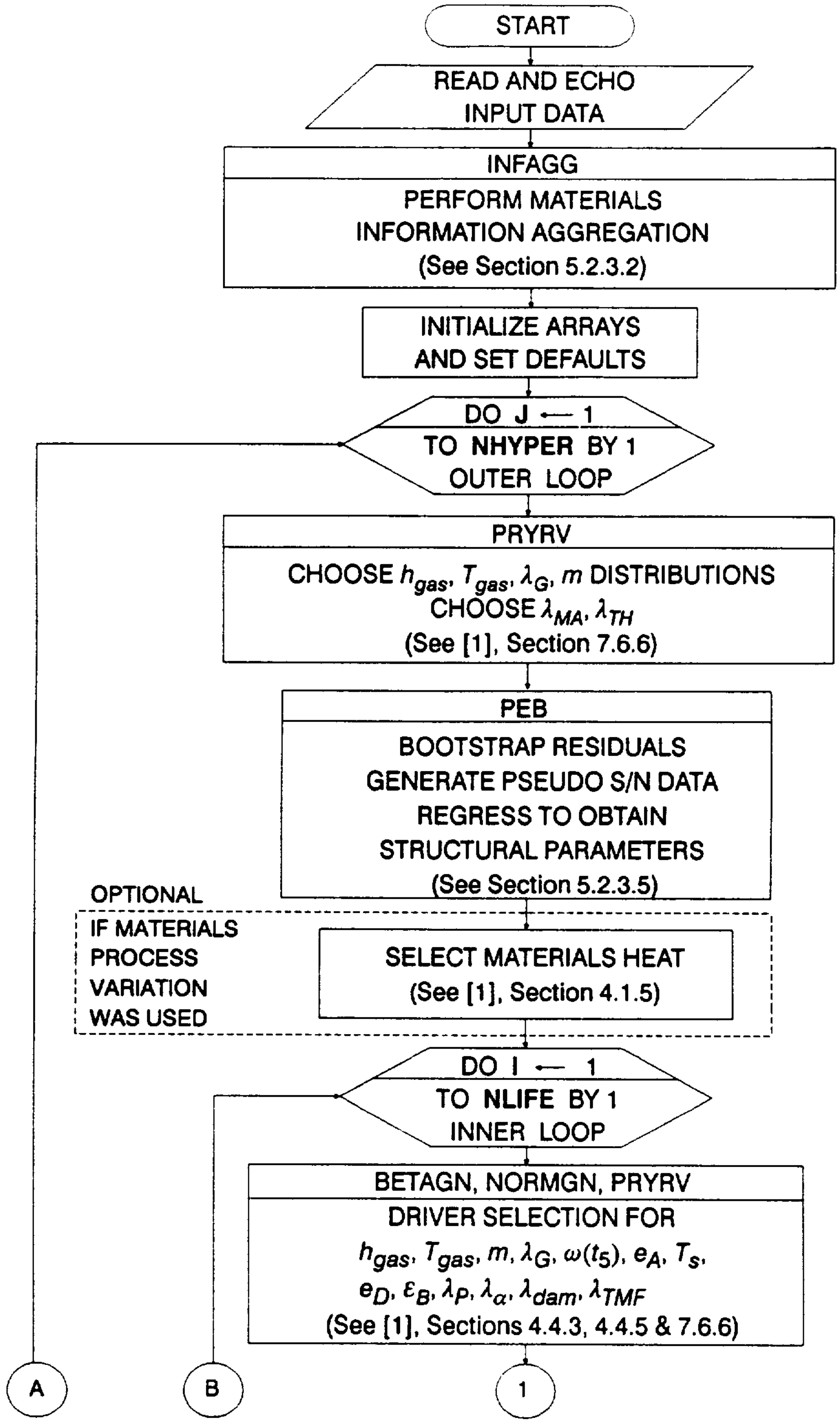


Figure 5.2-6 Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF Using the Nonparametric Materials Model

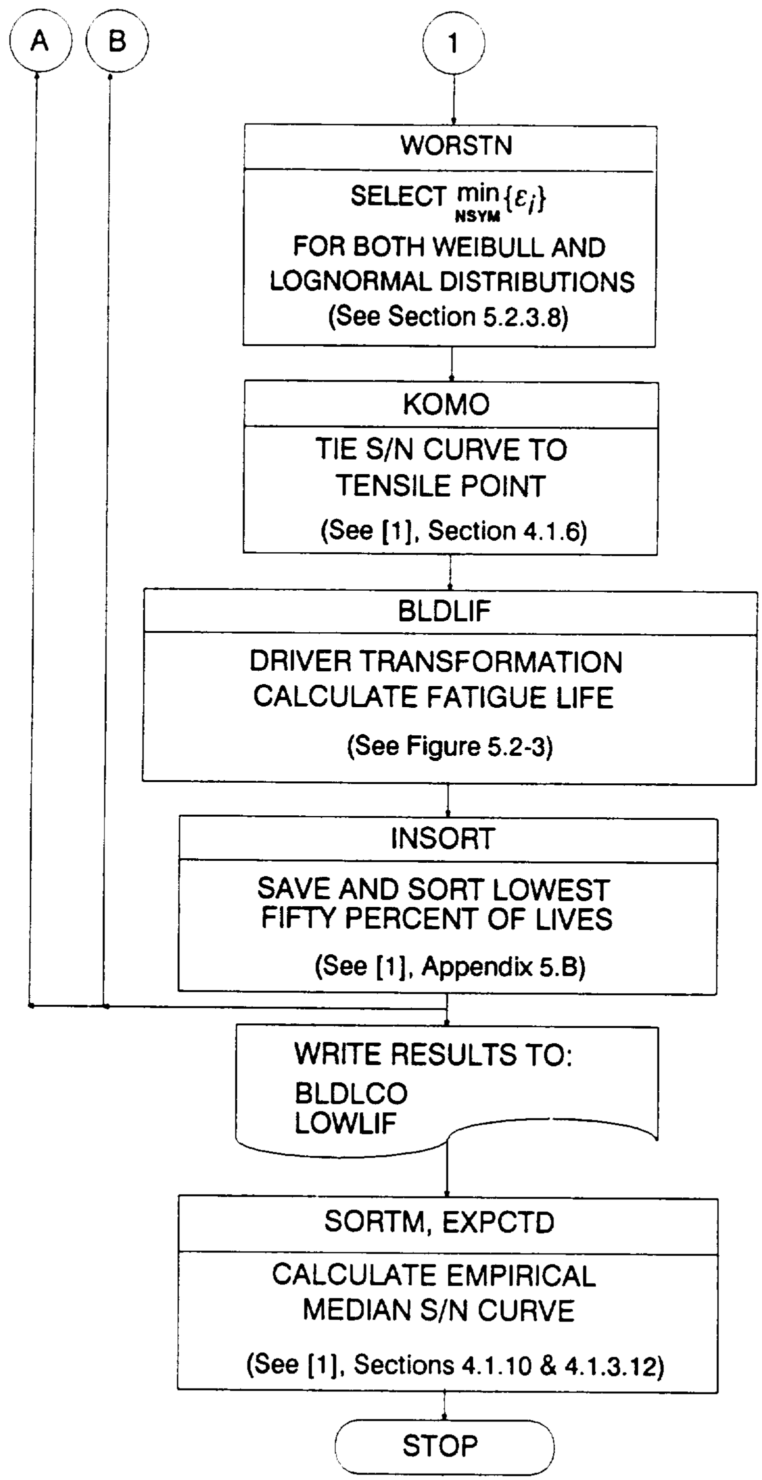


Figure 5.2-6 Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF Using the Nonparametric Materials Model (Cont'd)

A call to WORSTN provides the "worst of 50" materials intrinsic variability ε for both Weibull and Lognormal distributions. The routine WORSTN is described in Section 5.2.3.8.

When all the S/N model parameters have been selected for the region with S/N data, the S/N curve is tied to the tensile point S_0 by routine KOMO. The routine BLDLIF performs driver transformation and calculates the fatigue life. The flowchart for BLDLIF is given in Figure 5.2-3 and the routine is described below. Subprogram KOMO is described in [1], Sections 4.1.6.

The fatigue lives are arranged in ascending order in a list containing the lowest fifty percent of the lives. The INSORT routine performs an insertion sort with each new fatigue life. When the outer DO loop is completed, the list of lives representing the left-hand tail of the failure distribution is written to file LOWLIF. Subprogram INSORT is described in [1], Appendix 5.B.

The empirical median S/N curve is calculated next. The routine SORTM is called to sort the values of m and the routine EXPCTD calculates the median S/N curve. Sections 4.1.10 and 4.1.3.12 of [1] describe the routines SORTM and EXPCTD, respectively.

5.2.3.2 INFAGG Routine

The flowchart for the INFAGG routine is given in Figure 5.2-7. The routine controls the calls to the data input and information aggregation calculation routines. INFAGG starts by opening the following input and output files:[7]

NAME	TYPE	CONTENTS
RELATD	Input	Related material data input
RELATO	Output	Related material data echo

The arrays are then set to their default or initial values by routine INIT. Routine RCE reads the data from files SPECFD and RELATD, transforms (or converts) the stresses to an equivalent stress ratio of $R = -1.0$, and echoes the data to files SPECFO and RELATO. Routines INIT and RCE are described in [1], Sections 4.1.3.1 and 4.1.3.2.

The information aggregation begins with linear regression calculations performed by routine SW2SU2 on the combined specific and related data. Then the constraints on the shape parameters $\{m_j\}$ implied by the user-provided C_0 constraint are calculated by FINDMC. The routines SW2SU2 and FINDMC are

[7] The nonparametric model does not have the capability to utilize related data at this time.

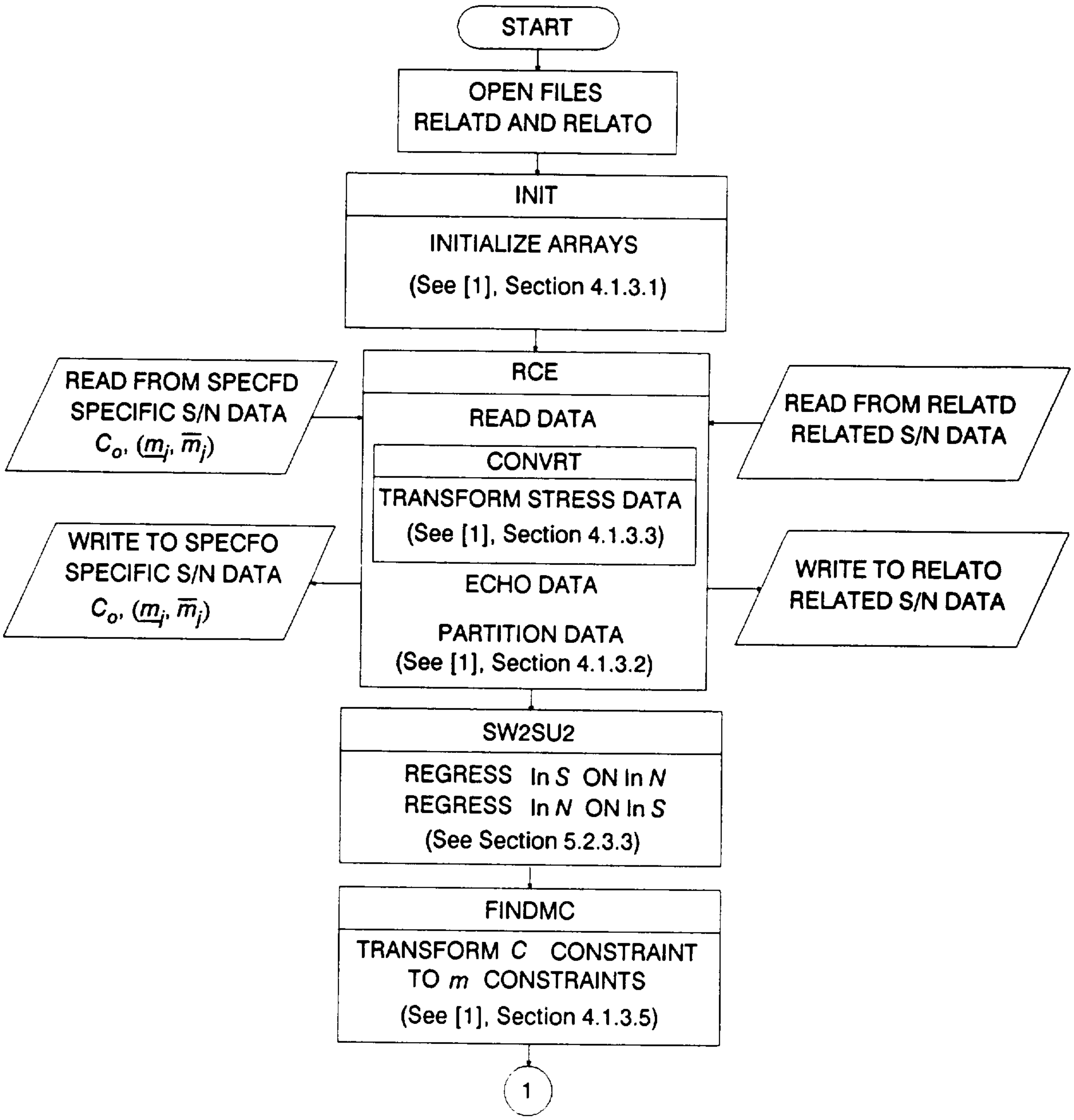


Figure 5.2-7 Flowchart for Subprogram INFAGG

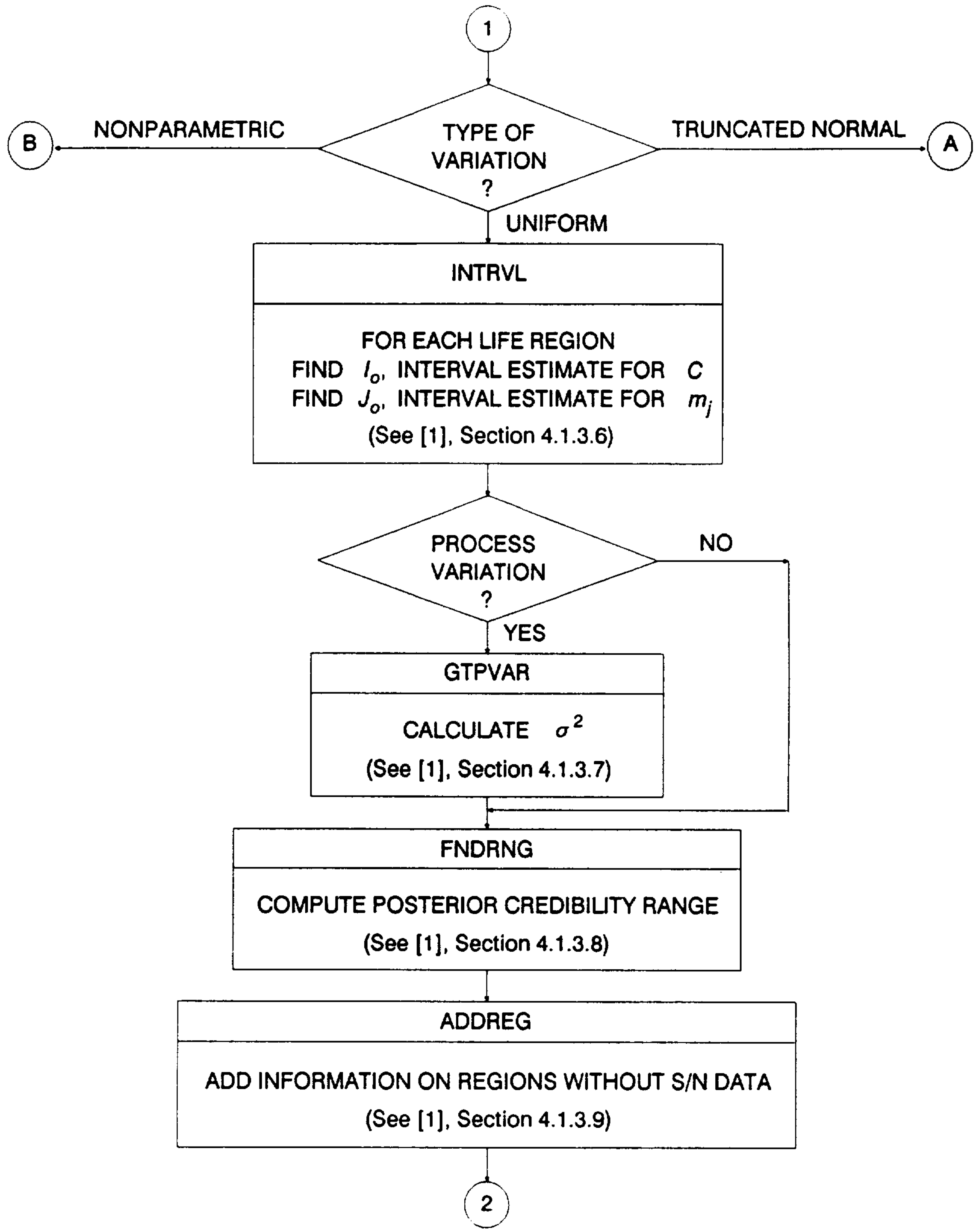


Figure 5.2-7 Flowchart for Subprogram INFAGG (Cont'd)

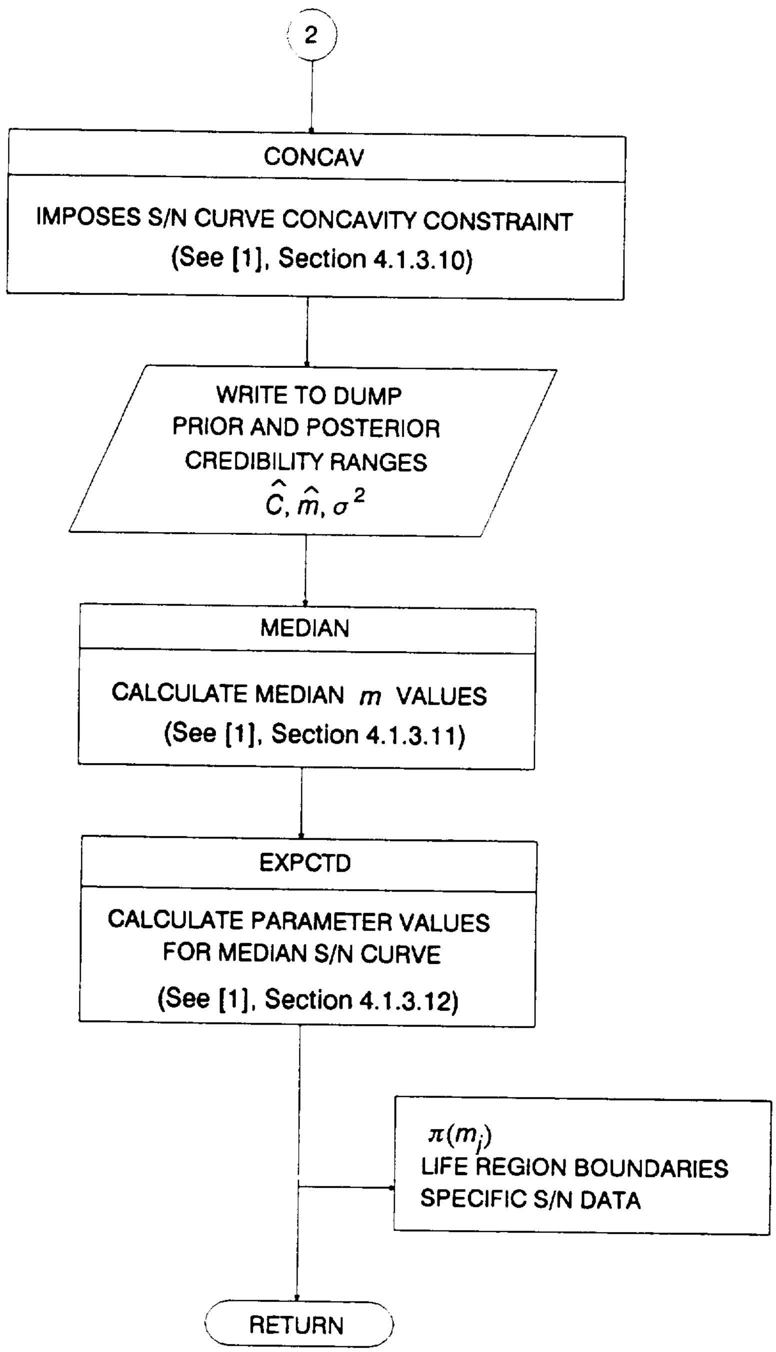


Figure 5.2-7 Flowchart for Subprogram INFAGG (Cont'd)

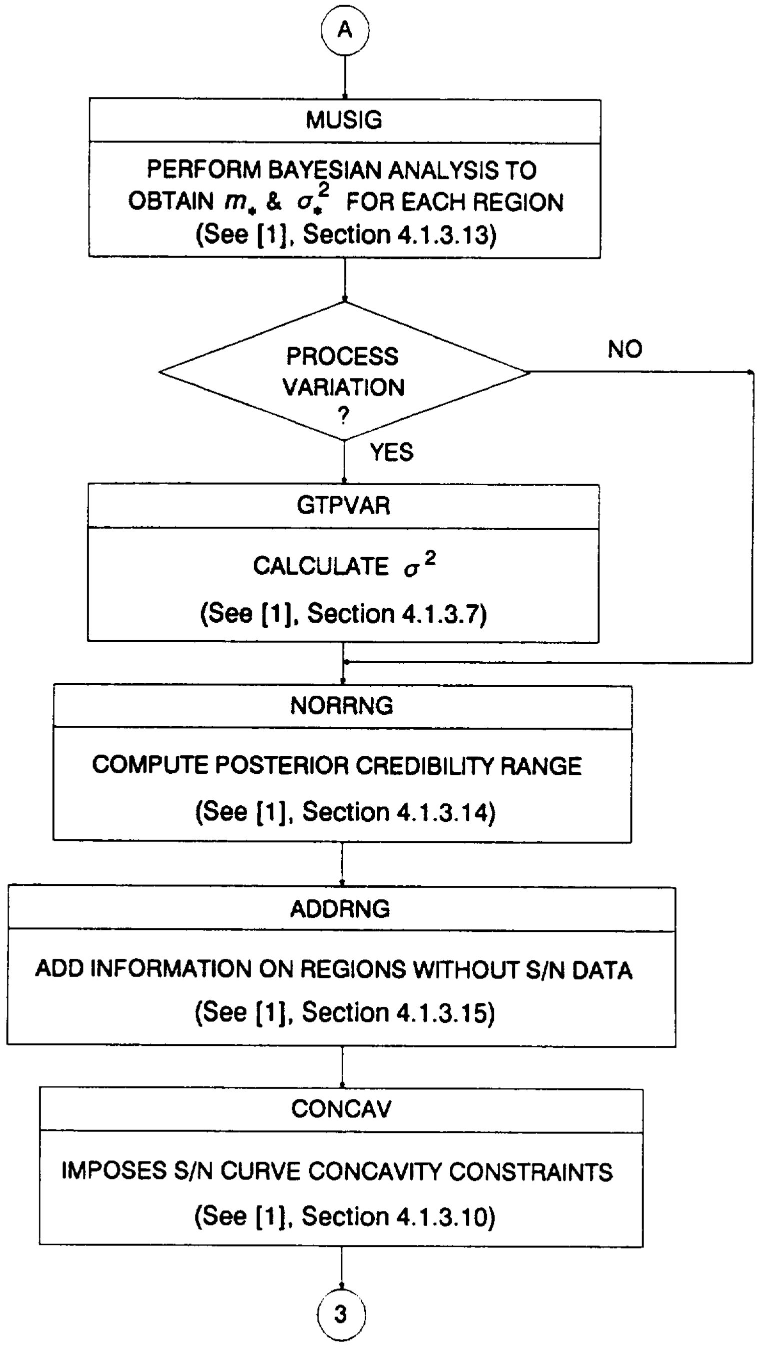
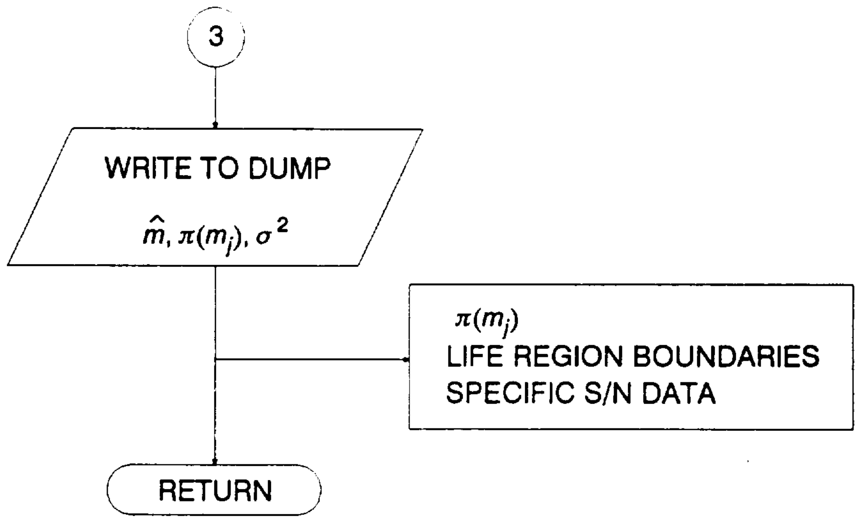


Figure 5.2-7 Flowchart for Subprogram INFAGG (Cont'd)



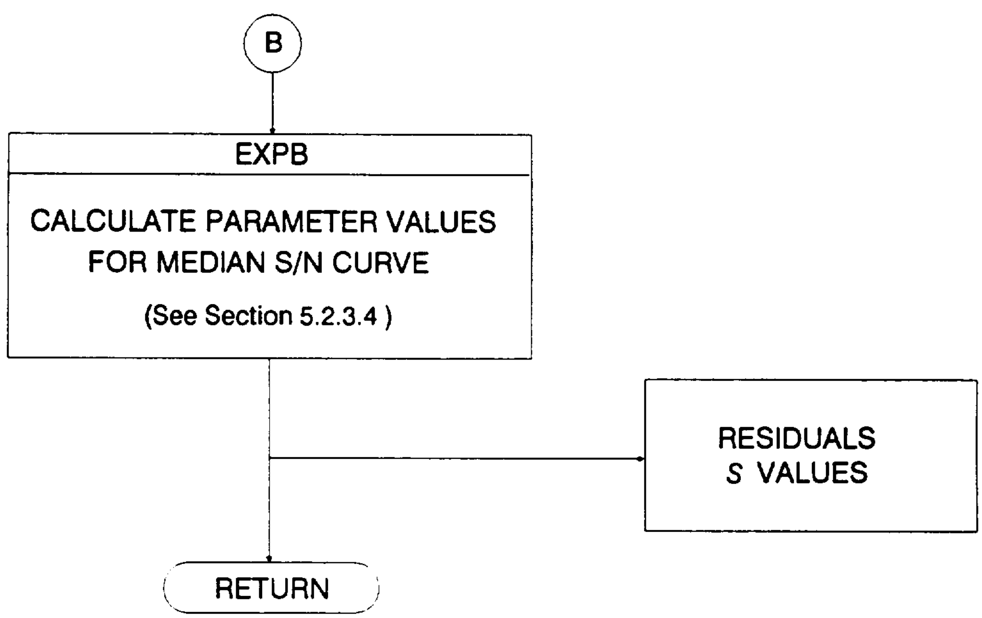


Figure 5.2-7 Flowchart for Subprogram INFAGG (Cont'd)

described in Section 5.2.3.3 and in [1], Section 4.1.3.5, respectively. The remaining routine calls depend upon the choice of distribution for the shape parameters.

The Uniform distribution case begins with the confidence interval calculations performed by INTRVL. By definition, the prior credibility ranges are the confidence intervals. If materials processes variation is specified, GTPVAR calculates σ^2, Equation 2-49 of [1], the extent of departures from the multiple heat median S/N curve warranted by the available information. The credibility ranges, C constraint, and the user-provided range information are combined by routine FNDRNG to obtain posterior credibility ranges on the shape parameters $\pi(m_j)$.[8] The user-supplied m ranges for the non-data life regions to the right of those with data are added to the array containing the $\pi(m_j)$ by routine ADDREG.[9] Concavity constraints are applied within subprogram CONCAV. The results of the calculations above are written to file DUMP. Finally, the median S/N curve is calculated. The median m's are found by MEDIAN and then used by EXPCTD to obtain the median curve parameters which are written to file DUMP. Routines INTRVL, GTPVAR, FNDRNG, ADDREG, CONCAV, MEDIAN, and EXPCTD are described in [1], Sections 4.1.3.6, 4.1.3.7, 4.1.3.8, 4.1.3.9, 4.1.3.10, 4.1.3.11, and 4.1.3.12, respectively.

The truncated Normal distribution case begins with the Bayesian analysis performed by MUSIG to find the Normal distribution parameters for the m's. If materials process variation is requested, GTPVAR calculates σ^2, the extent of departures from the multiple heat median S/N curve warranted by the available information, by using Equation 2-49 of [1]. The C constraint and the user-provided range information are combined by routine NORRNG to obtain posterior credibility ranges on the shape parameters $\pi(m_j)$.[8] The user-supplied m ranges and Normal distribution parameters for the non-data life regions to the right of those with data are added to the arrays containing the $\pi(m_j)$, m_*, and σ_*^2 by routine ADDRGN.[9] Concavity constraints are applied within subprogram CONCAV. Then results of the calculations above are written to file DUMP. Routines MUSIG, GTPVAR, NORRNG, ADDRGN, and CONCAV are described in [1], Sections 4.1.3.13, 4.1.3.7, 4.1.3.14, 4.1.3.15, and 4.1.3.10.

The bootstrapping option uses m and K estimates to obtain the median curve parameters using EXPB, which are then written to file DUMP. Routine EXPB is described in Section 5.2.3.4.

---

8 Combining information to obtain the posterior credibility ranges on m is discussed in [1], Page 2-13.

9 No data regions to the right are discussed in [1], Page 2-17.

5.2.3.3 SW2SU2 Routine

The flowchart for the SW2SU2 routine is given in Figure 5.2-8. The routine performs the y on x and x on y regressions to obtain the sample variances S_x^2, S_y^2, and S_{xy}, and the residual variances $S_{\hat{w}}^2$ and $S_{\hat{u}}^2$ for each life region. For the calculations, x is equal to ln S and y is equal to ln N. The routine SW2SU2 starts by initializing the arrays required for the calculations.

Within the outer region DO loop are two sets of nested DO loops, where the region counter $L = 1, \ldots, R$, and R is the number of life regions with S/N data.[10] In each set of DO loops, the outer loop is for each S/N data set, $j = 0, \ldots, P$, and the inner DO loop is for each data point in each data set, $k = 1, \ldots, N_j$. The first step is to calculate the sample means $\bar{x}_j$ and $\bar{y}_j$ for each data set in each region. Then the sample variances and degrees of freedom for each region in each data set are calculated as follows:

$$N\,S_x^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (x_{jk} - \bar{x}_j)^2$$

$$N\,S_y^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (y_{jk} - \bar{y}_j)^2$$

$$N\,S_{xy} = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (x_{jk} - \bar{x}_j)(y_{jk} - \bar{y}_j)$$

$$N = \sum_{j=0}^{P} (N_j - 1) - 1$$

where S_x^2, S_y^2, and S_{xy} are the sample variance of x, sample variance of y, and sample covariance of x and y, and N is the number of degrees of freedom for each life region, respectively. If S_{xy} is non-negative, the data does not support the analysis assumptions and the program run will be terminated. The sample variances are used to calculate the regression parameters d and b of Equations 2-20 and 2-21 of [1],

---

10 R is equal to one for the strain formulation.

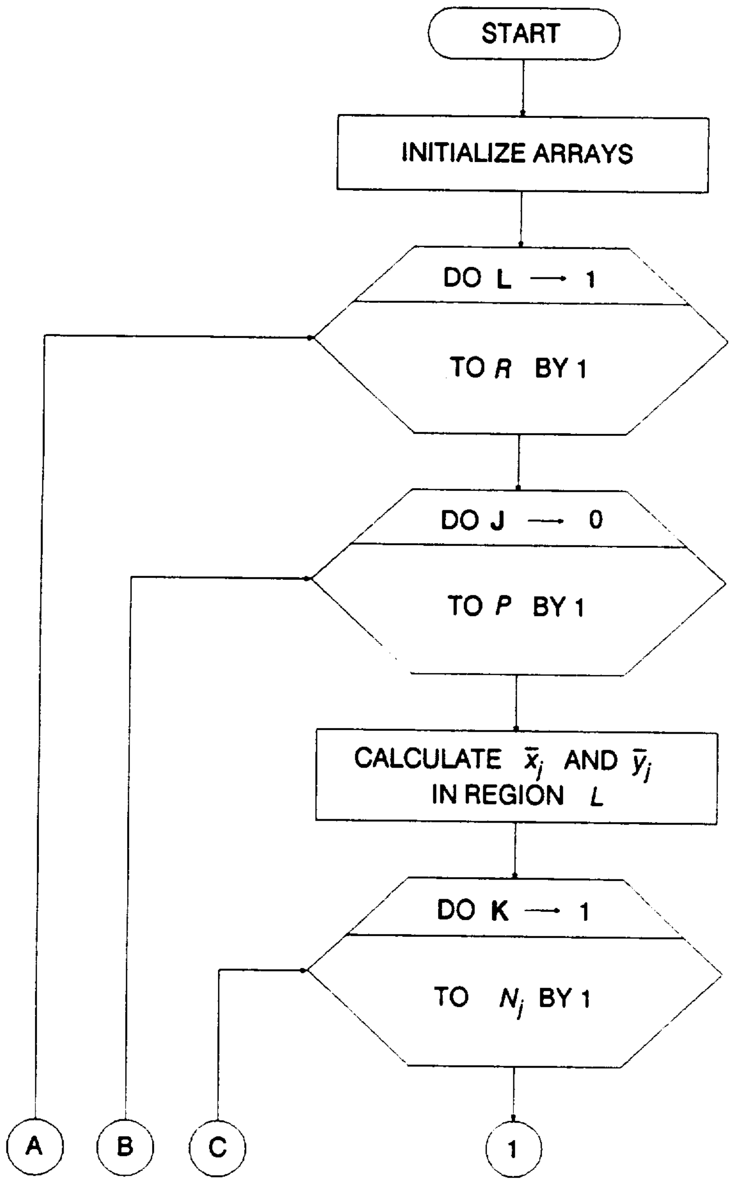


Figure 5.2-8 Flowchart for Subprogram SW2SU2

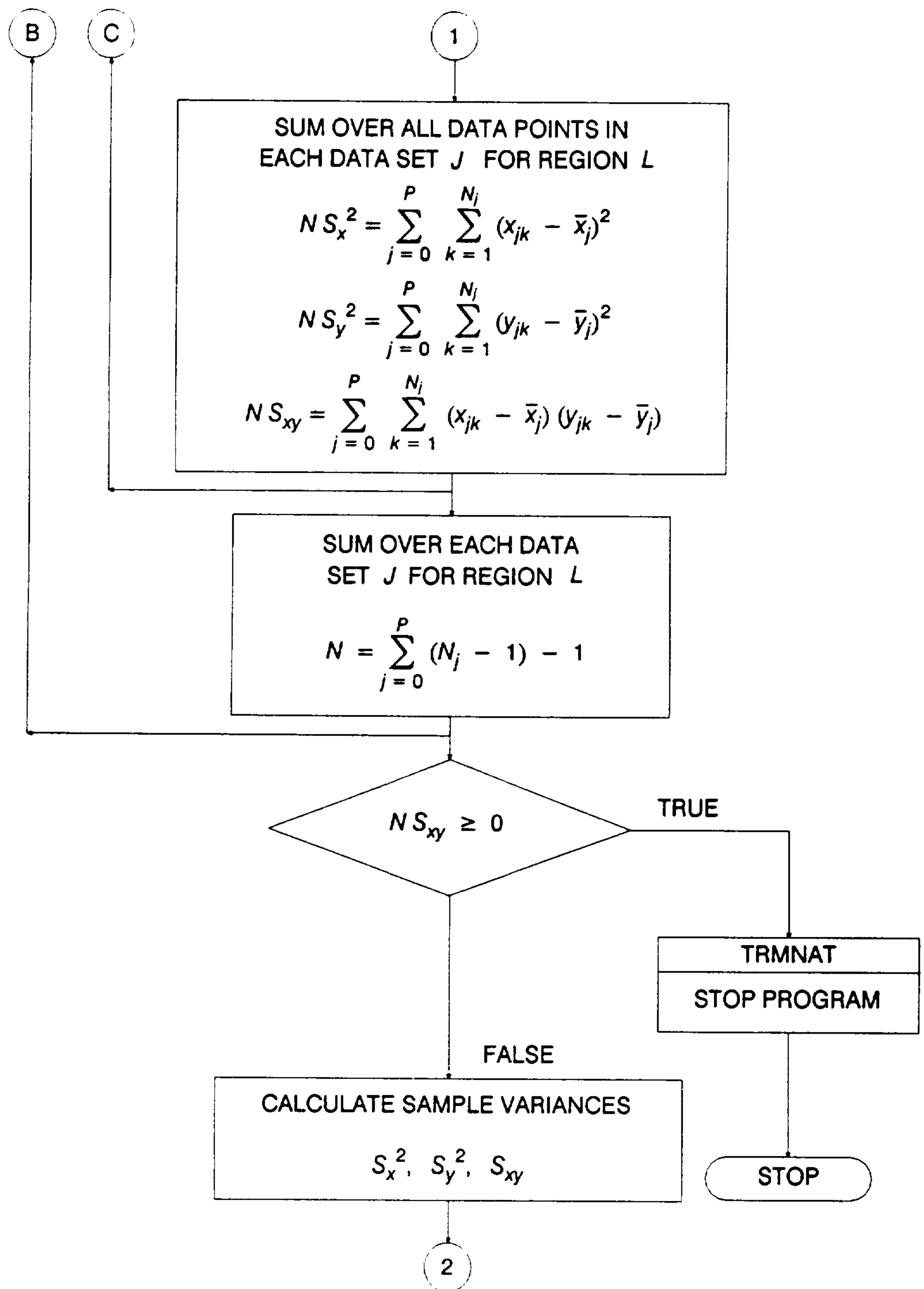


Figure 5.2-8 Flowchart for Subprogram SW2SU2 (Cont'd)

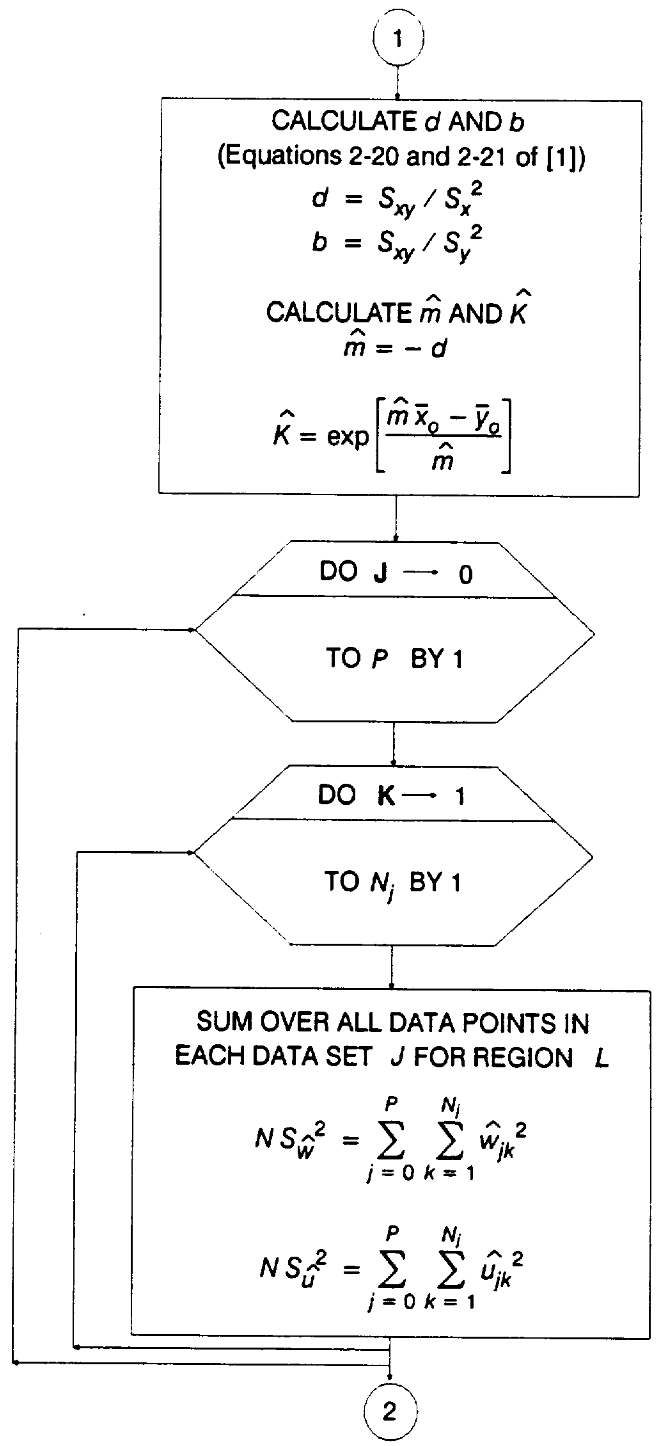


Figure 5.2-8 Flowchart for Subprogram SW2SU2 (Cont'd)

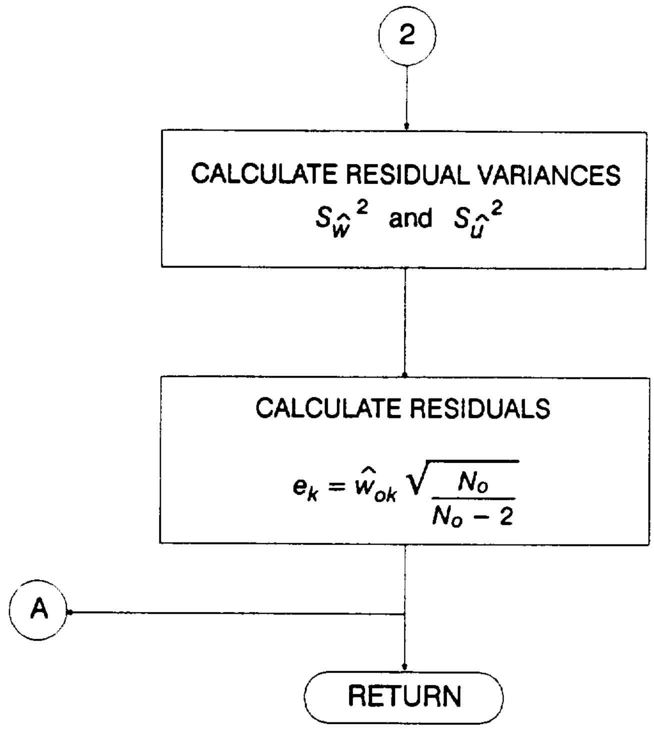


Figure 5.2-8 Flowchart for Subprogram SW2SU2 (Cont'd)

$$d = S_{xy} / S_x^2 \text{ and } b = S_{xy} / S_y^2$$

$$\hat{m} = -d \text{ and } \hat{K} = \exp\left[\frac{\hat{m}\,\bar{x}_o - \bar{y}_o}{\hat{m}}\right]$$

The second set of DO loops calculates the residuals $\underline{e}$ and the residual variances $S_{\hat{w}}^2$ and $S_{\hat{u}}^2$ for each life region given by

$$e_k = \hat{w}_{ok} \sqrt{\frac{N_o}{N_o - 2}}$$

$$N\,S_{\hat{w}}^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} \hat{w}_{jk}^2$$

$$N\,S_{\hat{u}}^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} \hat{u}_{jk}^2$$

where

$$\hat{w}_{jk} = (y_{jk} - \bar{y}_j) - d\,(x_{jk} - \bar{x}_j)$$

$$\hat{u}_{jk} = (x_{jk} - \bar{x}_j) - b\,(y_{jk} - \bar{y}_j)$$

from Equations 2-20 and 2-21 of [1].

5.2.3.4 EXPB Routine

The flowchart for the EXPB routine in given in Figure 5.2-9. The routine controls the calls to the median curve calculations for the bootstrap option. The routine uses the point estimates for the m and K to calculate the remainder of the parameters consistent with m, K, and the specific material data set. The stress values corresponding to the life region boundaries are obtained from FINDSB. If the tensile point S_o for the stress formulation is being used, then the S/N curve can be tied to S_o by routine KOMO.[11] The results of the calculations are written to file DUMP. Routines FINDSB and KOMO are described in [1], Sections 4.1.5.7 and 4.1.6, respectively.

[11] The tensile point calculations are included in routine EXPB in anticipation of future work on the bootstrap option.

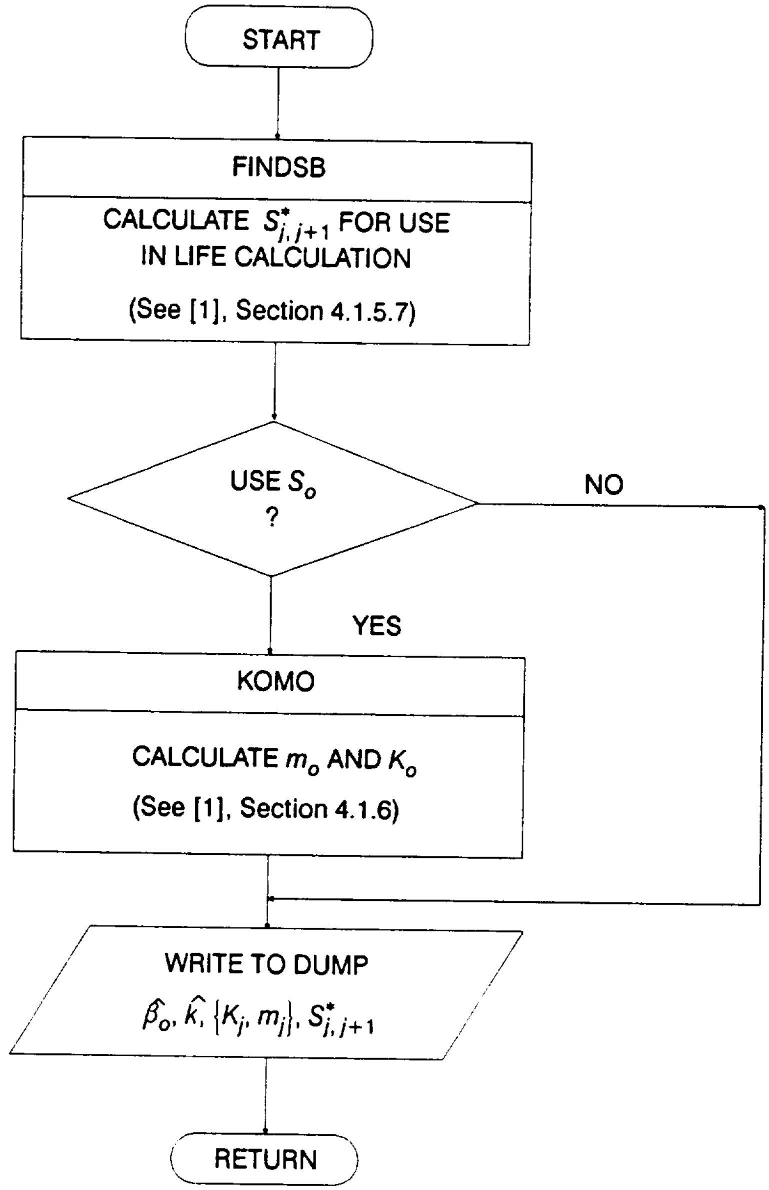


Figure 5.2-9 Flowchart for Subprogram EXPB

5.2.3.5 PEB Routine

The flowchart for the PEB routine is given in Figure 5.2-10. The routine controls the calls to the bootstrapping calculations. The calculations begin by the call to routine PICRES which performs the bootstrapping on the residuals and then generates the pseudo S/N data. Routine MREGR performs the regression to obtain a value of m for the pseudo S/N data. The routines PICRES and MREGR are described in Sections 5.2.3.6 and 5.2.3.7, respectively.

The remaining calculations find the $\{K_j\}$ and β_o parameters consistent with the pseudo S/N data and the calculated m. The calculations begin by routine TRNSFM transforming the specific material S/N data.[12] The transformation produces the $\{Z_i\}$ as a function of the S/N data, the m, and the life region boundary. Then, the sample mean and variance of Z are calculated by routine SMNVAR. KBETA computes the estimates of k and β_o. Then, the K is calculated by routine FINDK using Equations 2-37 through 2-41 of [1]. Finally, the stress value corresponding to the life region boundary is obtained from FINDSB. Routines TRNSFM, SMNVAR, KBETA, FINDK, and FINDSB are described in [1], Sections 4.1.5.3 through 4.1.5.7.

5.2.3.6 PICRES Routine

Routine PICRES bootstraps the residuals and generates the pseudo S/N data. The bootstrapping is performed by sampling with replacement on the set of residuals $\underline{e}$ for each stress value S_i. Then the pseudo S/N data is generated by calculating a new life value N_i^* for each stress value and selected residual e_i^* according to

$$N_i^* = \hat{a}\, S_i^{-\hat{m}}\, e_i^*$$

The inflation of the residuals by $\sqrt{\dfrac{N_o}{N_o - 2}}$ was performed in routine SW2SU2.

5.2.3.7 MREGR Routine

The flowchart for the MREGR routine is given in Figure 5.2-11. The routine performs the y on x and x on y regressions to obtain the estimate of the shape parameter m. For the calculations, x is equal to ln S and y is equal to ln N. MREGR starts by initializing the arrays required for the calculations.

---

[12] The S/N data transformation is discussed in [1], Page 2-16.

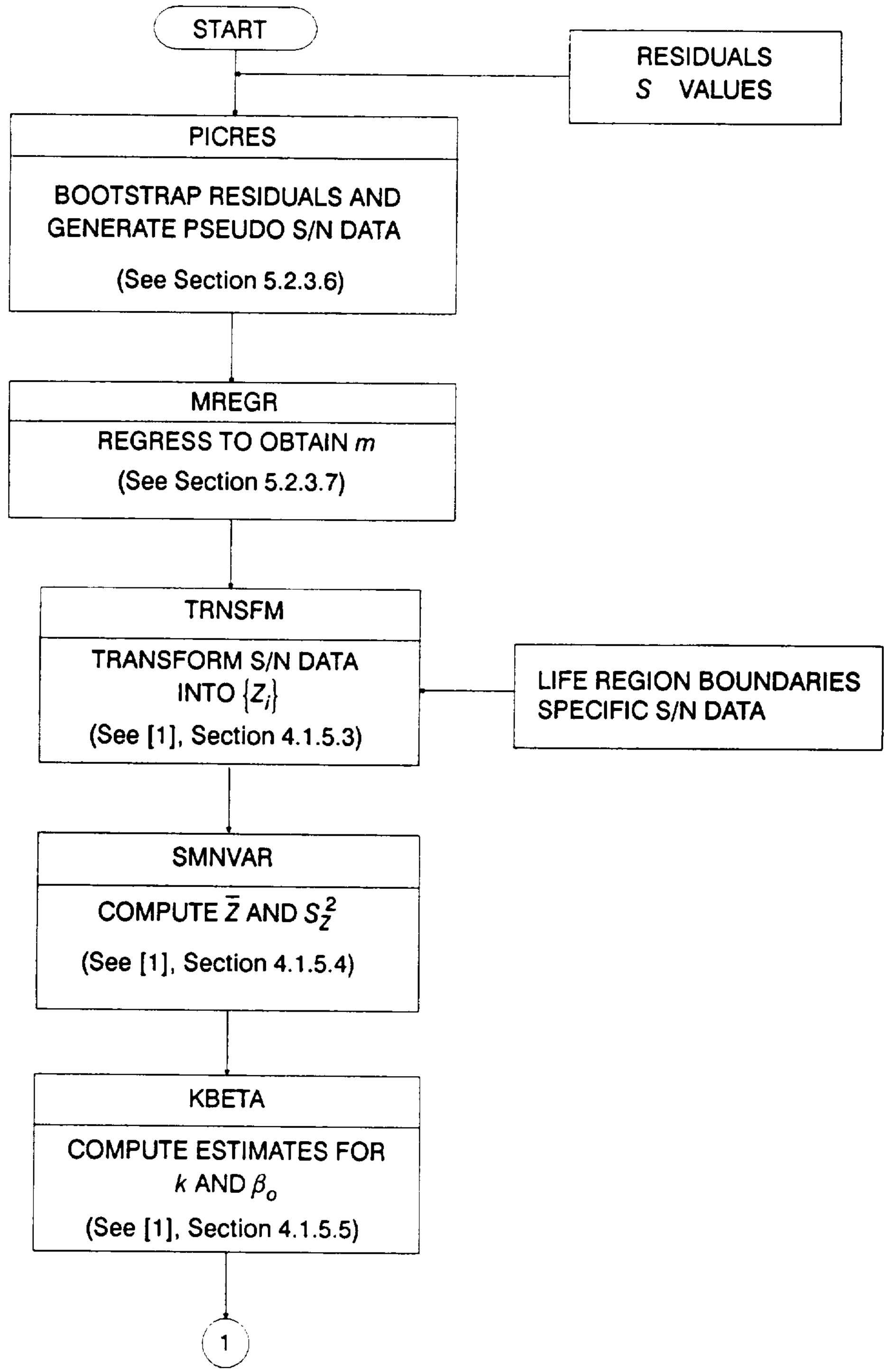


Figure 5.2-10 Flowchart for Subprogram PEB

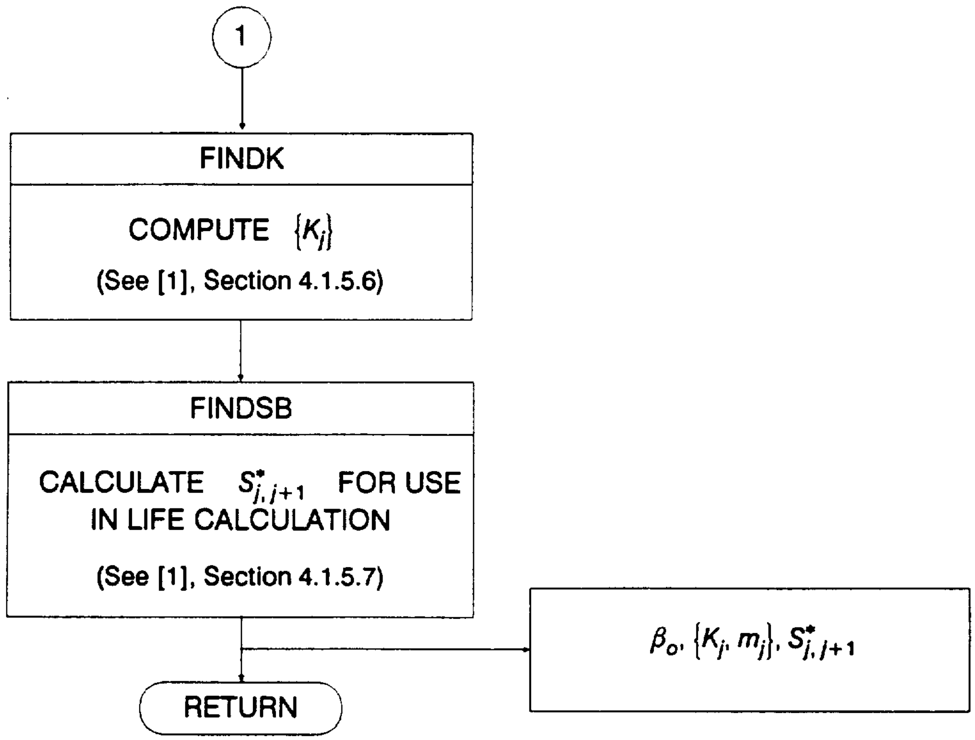


Figure 5.2-10 Flowchart for Subprogram PEB (Cont'd)

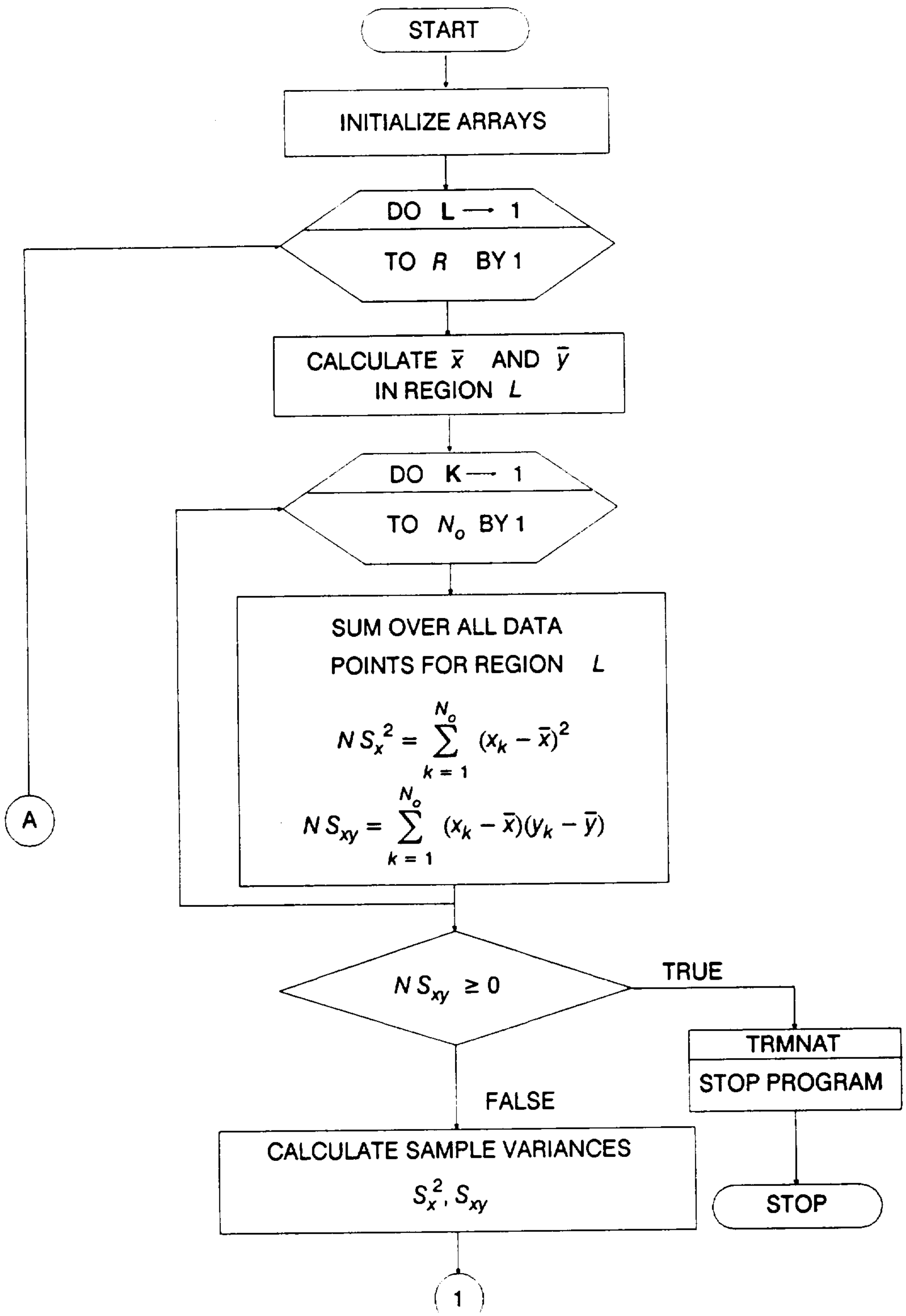


Figure 5.2-11 Flowchart for Subprogram MREGR

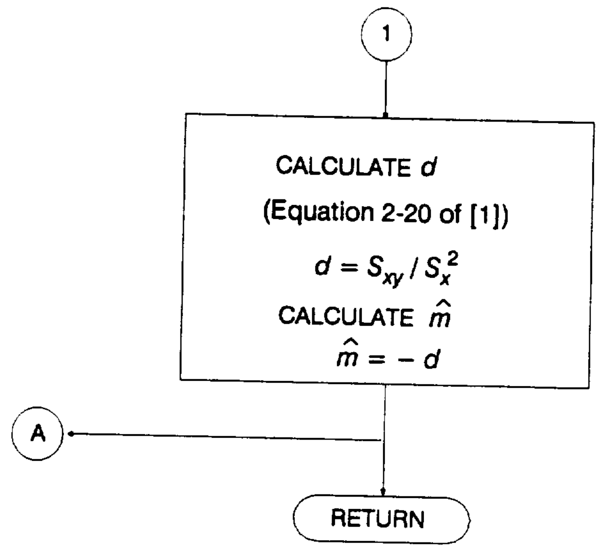


Figure 5.2-11 Flowchart for Subprogram MREGR (Cont'd)

Within the outer region DO loop are two inner DO loops, where the region counter $L = 1, ..., R$, is the number of life regions with S/N data.[13] Each inner DO loop is for each data point, $k = 1, ..., N_o$. The first step is to calculate the sample means $\bar{x}$ and $\bar{y}$ in each region. Then the sample variances for each region are calculated as follows:

$$N\, S_x^2 = \sum_{j=1}^{N_o} (x_j - \bar{x})^2$$

$$N\, S_{xy} = \sum_{j=1}^{N_o} (x_j - \bar{x})(y_j - \bar{y})$$

where S_x^2 and S_{xy} are the sample variance of x and the sample covariance of x and y for each region, respectively. If S_{xy} is non-negative, the data does not support the analysis assumptions and the program run will be terminated. The sample variances are used to calculate the regression parameter d of [1], Equation 2-20,

$$d = S_{xy} \,/\, S_x^2.$$

Then the shape parameter m is given by

$$m = -\, d.$$

5.2.3.8 WORSTN Routine

The following routine can be used to provide an analytic solution to the problem of selecting the smallest of N lives for either the parametric or bootstrapping characterization of materials model specification error.

Routine WORSTN performs the worst of N selection of the materials intrinsic variation parameter ε described in Section 3.2.7.3. The first step is to obtain a Uniform(0,1) random variate for F. Then the Weibull worst of N variate is given by

$$\varepsilon = \exp\left[\left(\ln\left(\frac{-\ln(1-F)}{N}\right) - \ln(\ln 2)\right)\frac{m}{\beta_o}\right]$$

Finally the Lognormal worst of N variate is obtained using the algorithm given in 26.2.23 of [4].

---

[13] R is currently equal to 1 for the bootstrapping option, but the region DO loop has been included in anticipation of future work on the bootstrap option.

Section 5.3
High Cycle Fatigue Analysis Software

5.3.1 Introduction

This section presents a description of the computer program which implements the HCF analysis discussed in Section 4. The code for analyzing the ATD-HPOTP first and third stage turbine blades is described below in Section 5.3.2. The overall layout of the program is described by using a main flowchart that refers to other flowcharts, which describe subprograms and key portions of the main program in greater detail. The program tree structure, a list of subprograms, a description of the key variables, and the FORTRAN source listing for the HCF analysis code BLDHCF are given in Section 7.3. The materials characterization subprograms and those subprograms that are of a generic nature, such as the random variate generators, are described in [1], Section 4.1 and Section 4.4 respectively. A glossary of standard flowchart symbols is given for the reader's benefit in Appendix 5.A.

5.3.2 BLDHCF Program

The HCF analysis of the ATD-HPOTP first and third stage turbine blades is implemented as the FORTRAN program BLDHCF. Figure 5.3-1 shows the structure of the PFM for the Blade. This section provides the description and flowcharts for program BLDHCF.

5.3.2.1 Main Routine

The master flowchart for the BLDHCF program is given in Figure 5.3-2. The program starts by opening the following input and output files:[14]

NAME	TYPE	CONTENTS
BLDHCD	Input	Analysis data
BLDHCO	Output	Input data echo, results
RELATD	Input	Related material data input
RELATO	Output	Echo of information in RELATD
DUMP	Output	Results of materials characterization calculations
IOUTPR	Output	Run information and user-requested information
LOWLIF	Output	First one percent of sorted fatigue lives

Routine DRVRIN is called to read the input data from the BLDHCD file. An echo of the input data is written onto BLDHCO. DRVRIN is described in Section 5.3.2.2. The related material S/N information is read from the file RELATD and processed in the INFAGG routine. INFAGG controls the materials information aggregation and

[14] Files RELATD and RELATO are opened in INFAGG.

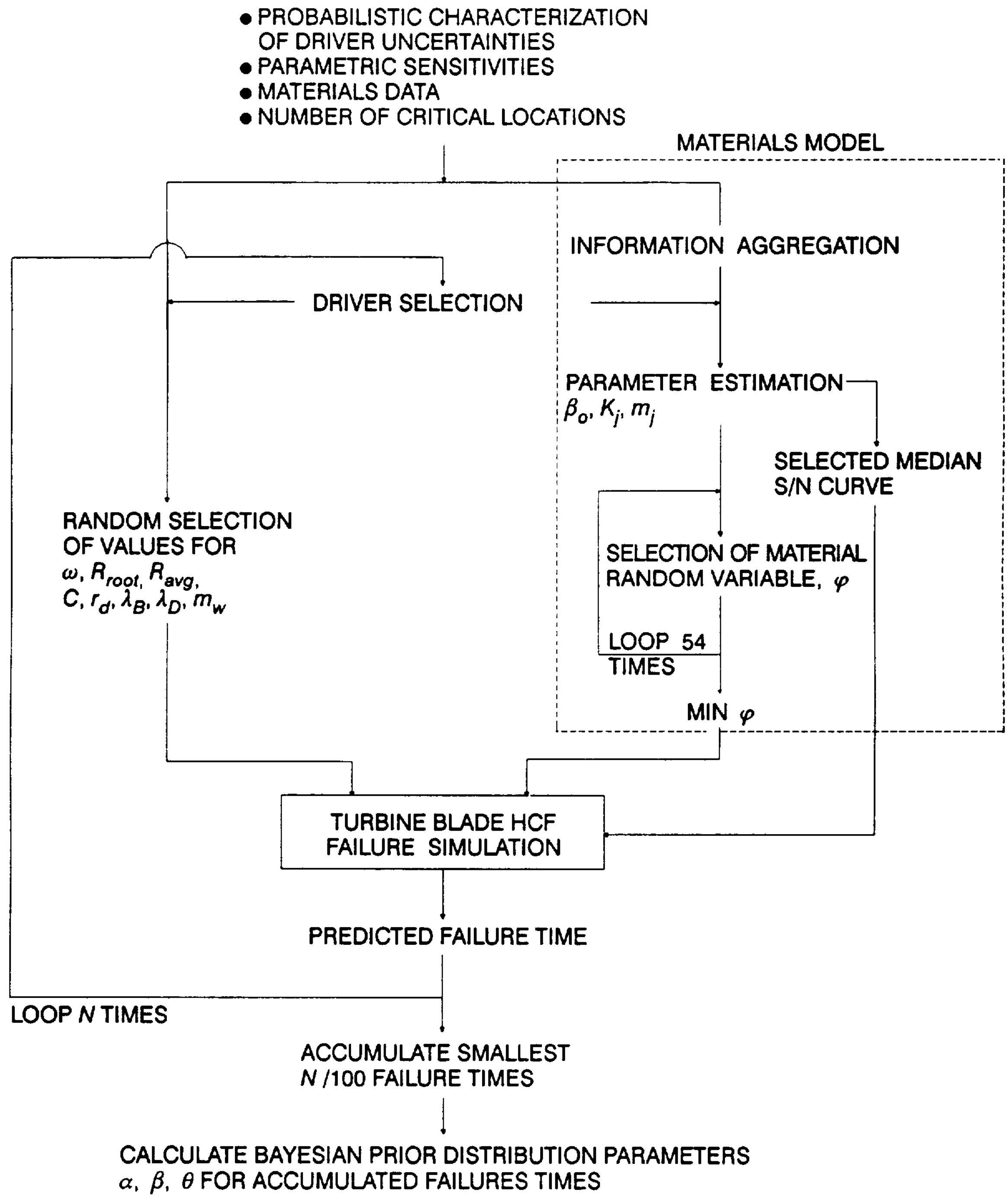


Figure 5.3-1 Structure of the Turbine Blade HCF Probabilistic Failure Model

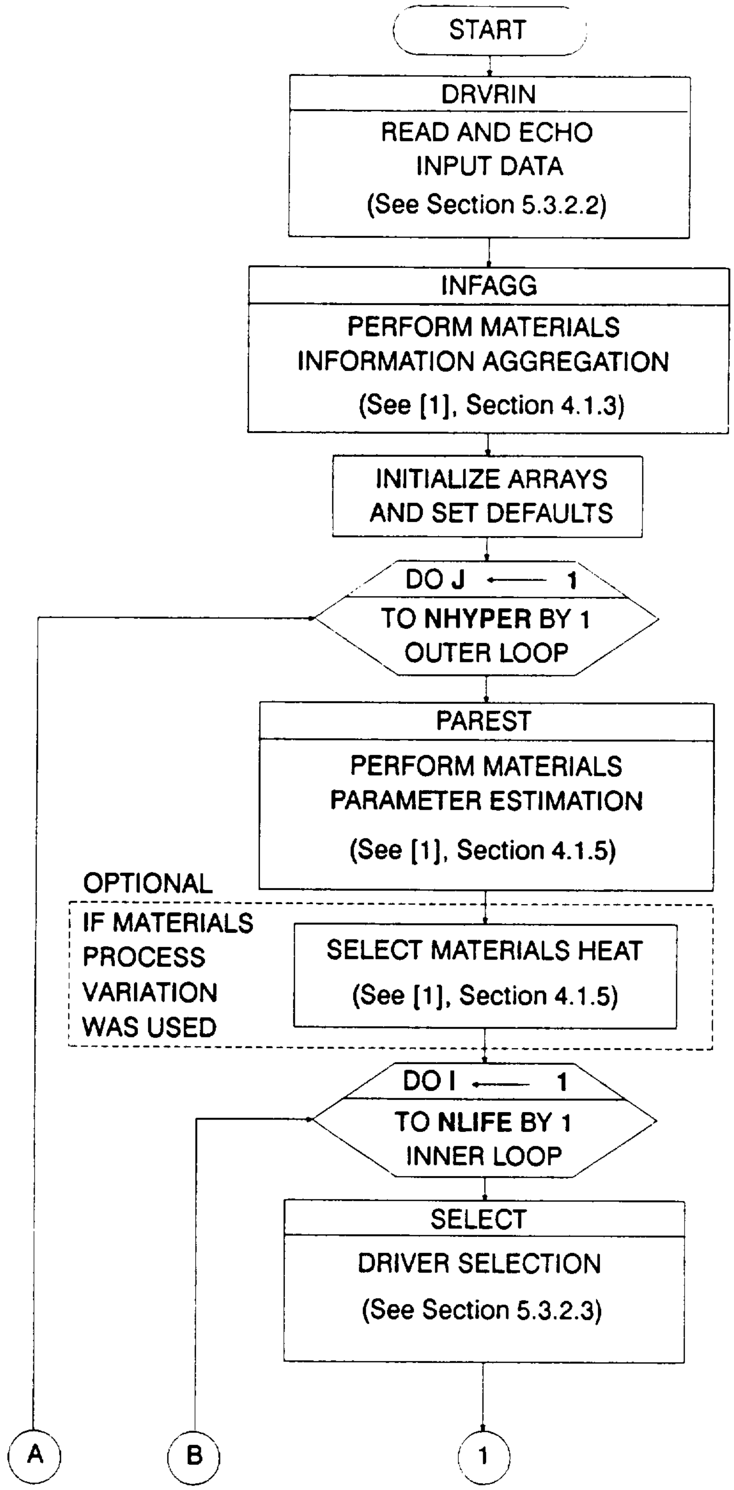


Figure 5.3-2 Main Flowchart for the ATD Blade HCF Analysis Program BLDHCF

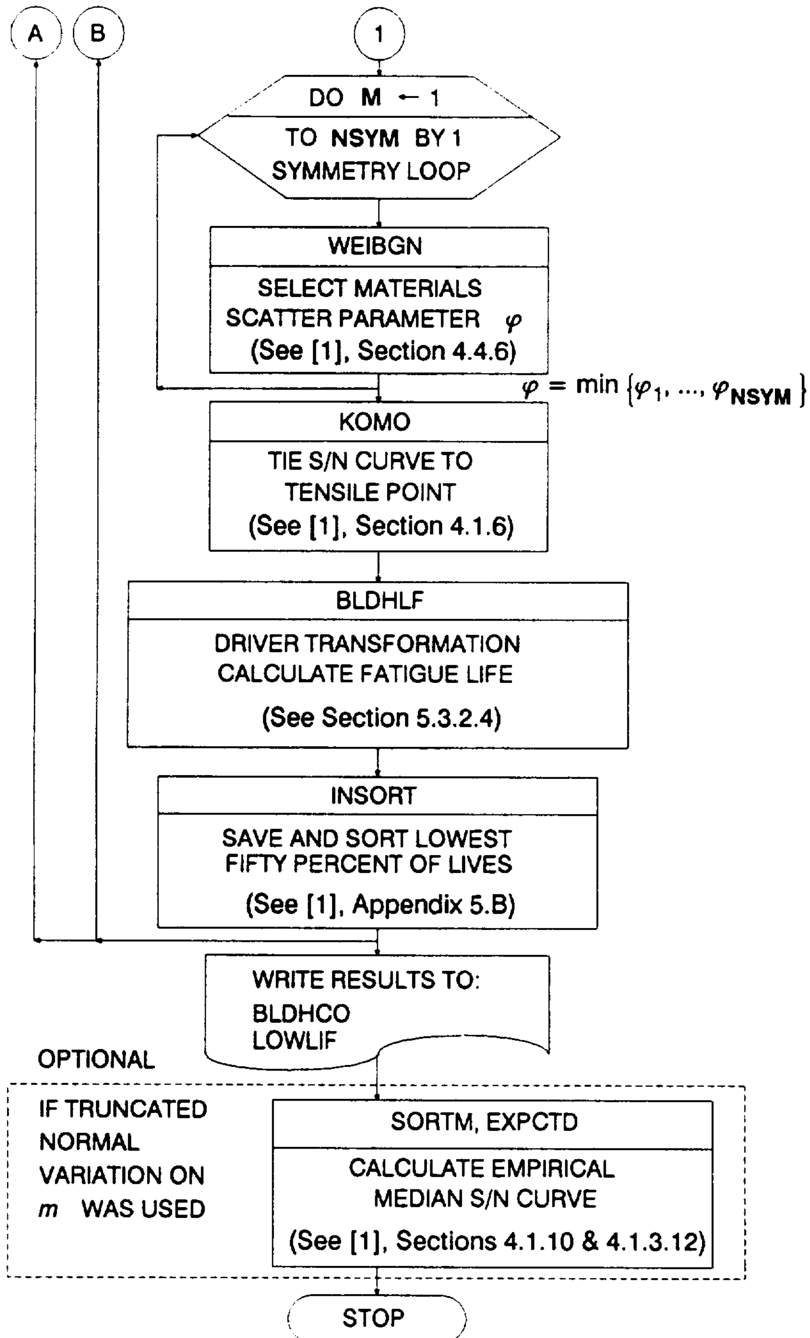


Figure 5.3-2 Main Flowchart for the ATD Blade HCF Analysis Program BLDHCF (Cont'd)

is described in [1], Section 4.1.3. The arrays and variables are then set to their default or initial values.

In the outer DO loop of the simulation, the PAREST routine controls the calculations for estimating the parameters for the S/N model. Routine PAREST is described in [1], Section 4.1.5. If materials process variation is included, the materials parameter Z in [1], Equation 2-48 is selected by calling the NORMGN routine and then transforming the resulting Normal variate to a Lognormal variate.

The inner DO loop for the simulation performs the driver selection. The SELECT routine controls the driver selection and is described in Section 5.3.2.3.

In the symmetry DO loop, the materials model parameter φ is found from the minimum of 54 draws of a Weibull distribution. Calls to WEIBGN provide the 54 values of φ. Subroutine WEIBGN is described in [1], Section 4.4.6.

When all the S/N model parameters have been selected for the region with S/N data, the S/N curve is tied to the tensile point S_o by routine KOMO. The routine BLDHLF performs driver transformation and calculates the fatigue life. The BLDHLF routine is described in Section 5.3.2.4. Subprogram KOMO is described in [1], Sections 4.1.6.

The fatigue lives are arranged in ascending order in a list containing the lowest fifty percent of the lives. The INSORT routine performs an insertion sort with each new fatigue life. When the outer DO loop is completed, the list of lives representing the left-hand tail of the failure distribution is written to file LOWLIF. Subprogram INSORT is described in [1], Appendix 5.B.

If a truncated Normal distribution was used for the materials shape parameter m, the empirical median S/N curve will be calculated upon user request. The routine SORTM is called to sort the values of m and the routine EXPCTD calculates the median S/N curve. Sections 4.1.10 and 4.1.3.12 of [1] describe the routines SORTM and EXPCTD, respectively.

5.3.2.2 DRVRIN Routine

The DRVRIN routine controls the input/output of the driver distributions and the structural and geometric parameters. The input data is read from file BLDHCD and the data is written to file BLDHCO.

5.3.2.3 SELECT Routine

The SELECT routine controls the driver selection. The drivers ω, R_{root}, R_{avg}, C, r_d, λ_B, λ_D, and m_w are selected by calling NORMGN and PRYRV which draw from

Normal and Uniform distributions respectively. The random variate routines NORMGN and PRYRV are described in [1], Sections 4.4 and 7.6.

5.3.2.4 BLDHLF Routine

BLDHLF performs the driver transformation and fatigue life calculation. The flowchart for the driver transformation is given in Figure 5.3-3. First, the mass flow rate $\dot{m}$ and the change in enthalpy Δh are calculated using the performance balance characterization. Next, the blade root mean stress $\overline{\sigma_{BR}}$ calculation is performed, Equation 4-1. The blade undamped vibratory stress σ_{UD} is calculated based on the empirical characterization as a function of $\overline{\sigma_{BR}}$. The blade damper effectiveness characterization model is used to obtain the ratio of damped vibratory stress to undamped vibratory stress σ_D / σ_{UD} as a function of the centrifugal force produced by the blade damper. Then the mean and alternating stresses, the maximum and minimum stresses, and the stress ratio are calculated using Equations 4-2 through 4-6.

The flowchart for the fatigue life calculation is given in Figure 5.3-4. First, the equivalent zero mean stress is calculated using the Walker relation of Equation 4-7. The life in cycles N_f corresponding to the equivalent stress cycle is obtained from the S/N curve by calling the GTLIFE routine. The GTLIFE routine is described under materials characterization in [1], Section 4.1.8. The failure life in seconds L is calculated as a function of N_f, the rotor speed ω, and the number of stators N_s.

References

[1] Moore, N., et al., An Improved Approach for Flight Readiness Certification – Methodology for Failure Risk Assessment and Application Examples, JPL Publication 92-15, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, June 1, 1992.

[2] Fatigue Crack Growth Computer Program "NASA/FLAGRO" Manual, NASA-JSC 22287, 1986.

[3] Broek, D., Elementary Engineering Fracture Mechanics, Martinus Nijhoff Publishers, Dordrecht, The Netherlands, 1986.

[4] Abramowitz, M., and Stegun, I. A., editors, Handbook of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series 55, Issued June 1964, Ninth Printing, November 1970 with corrections.

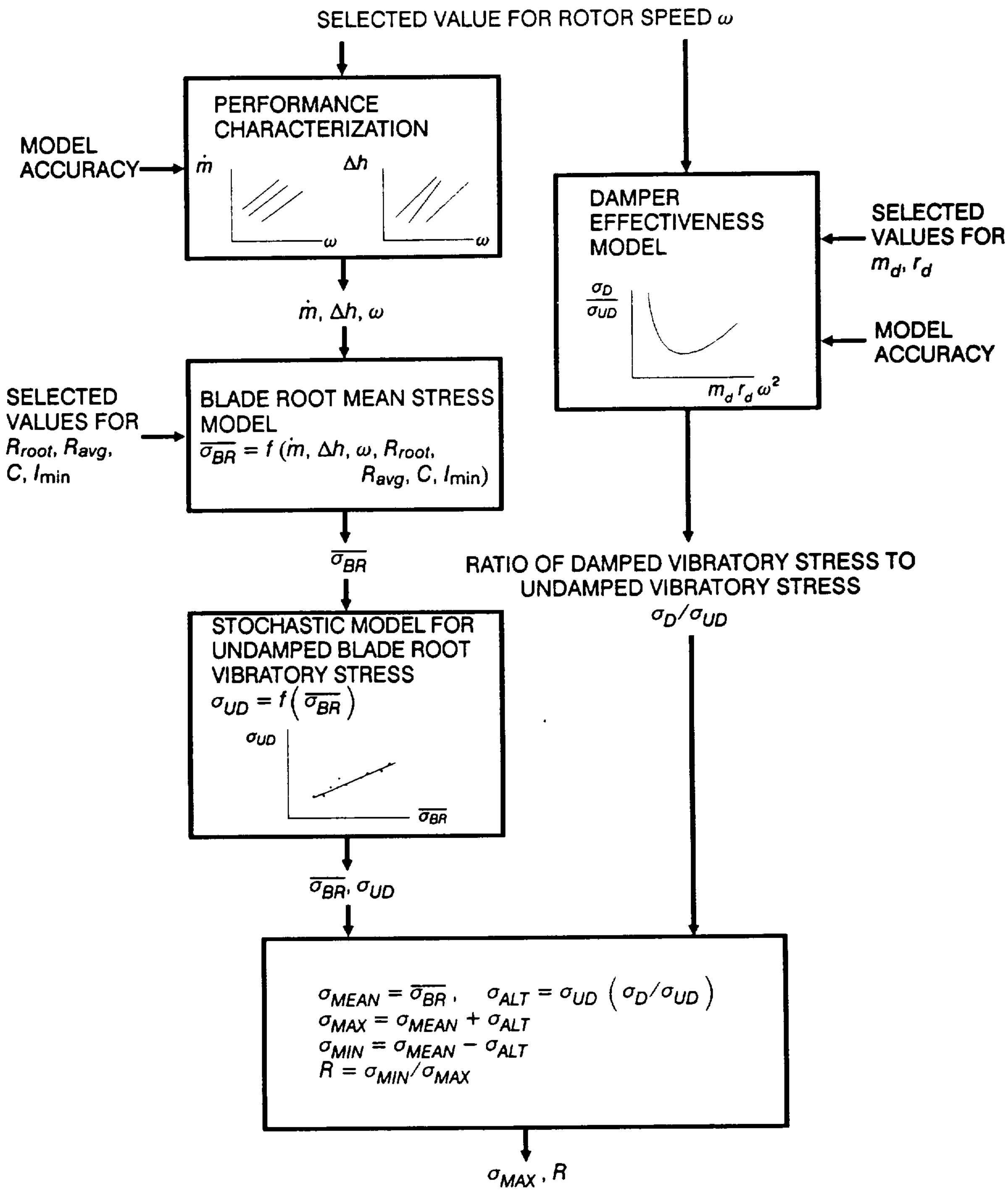


Figure 5.3-3 Structure of the Driver Transformation for the Turbine Blade HCF Model

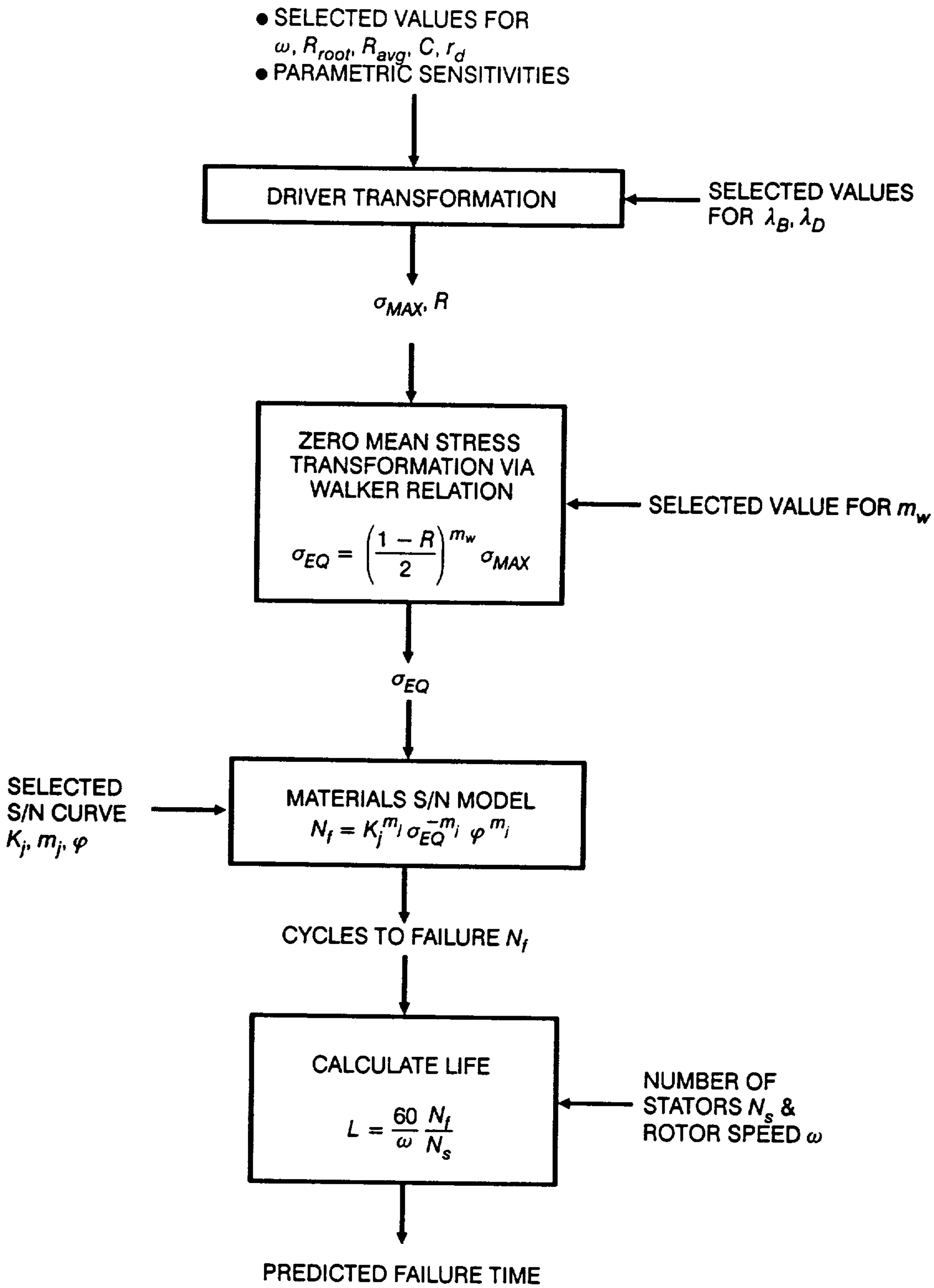


Figure 5.3-4 Structure of the Failure Life Calculation for Turbine Blade HCF

Appendix 5.A
Program Flowchart Symbols

The symbols employed in the flowcharts are given in Figure 5.A-1.

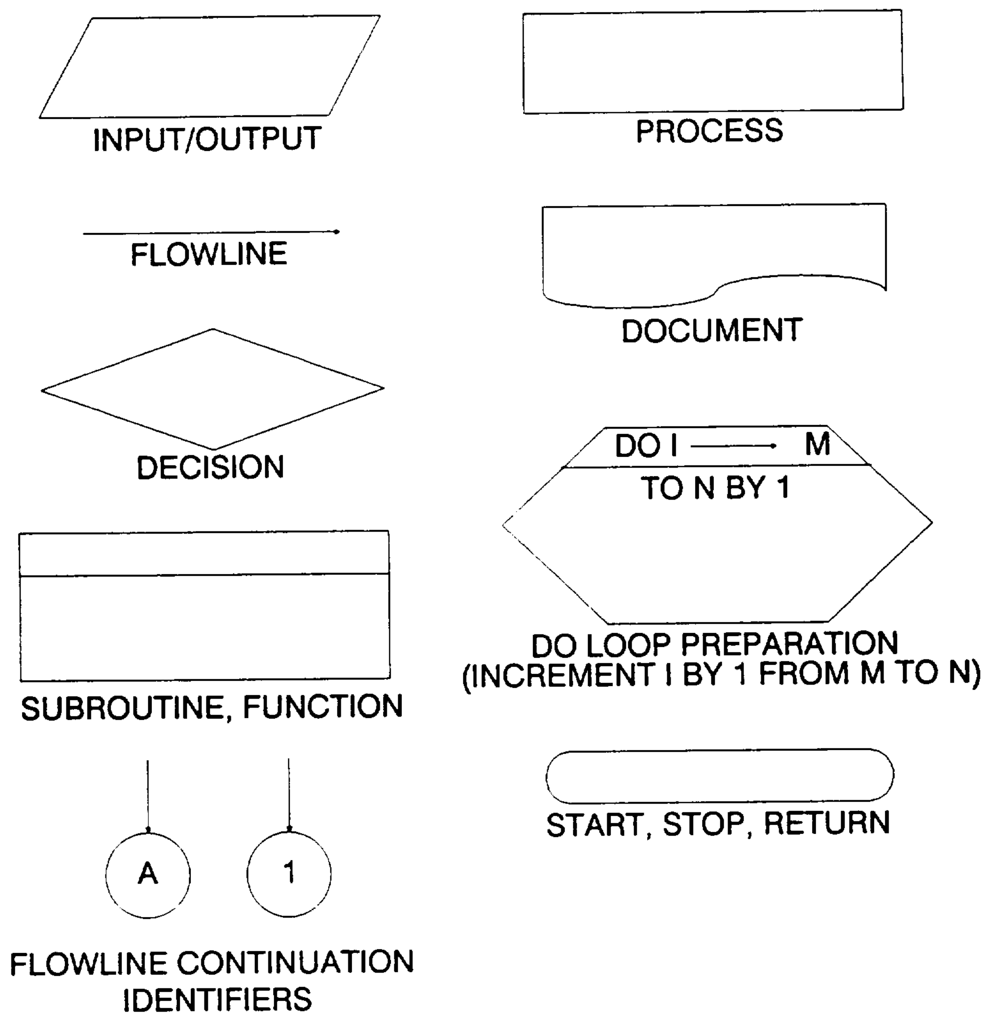


Figure 5.A-1 Program Flowchart Symbols

6.0 Software User's Documentation

Section 6.1

Crack Growth Analysis User's Guide

6.1.1 PROCRK Program

A user's guide for running the crack growth analysis code PROCRK is given here. The crack growth analysis methodology is discussed in Section 2.2, the program description and flowcharts are presented in Section 5.1, and the code structure and listing are provided in Section 7.1.

The PROCRK program was used to analyze the crack growth failure of the HPOTP heat exchanger (HEX) coil and the proposed external heat exchanger (EXHEX). The dynamic load input for the program consists of narrow-band and sinusoidal reference time histories. These reference time histories are generated using the program NBSIN. The output of PROCRK includes the simulated B-lives and a list of the lowest one percent of lives. The list of lives may be used as input to the regression programs of Section 4.2 in [1] in order to compute the parameters of the Bayesian prior failure distribution. This prior distribution and the success/failure data are used as input to the Bayesian updating program BAYES to obtain a posterior failure distribution. Programs NBSIN and BAYES are described in Sections 4.5 and 4.3, respectively, of [1].

6.1.2 How to Use Program PROCRK

The program PROCRK is intended to be run in batch (i.e., background) mode. PROCRK requires the input file CRKDAT and a set of load data files containing the reference time histories. The names of the load data files must be defined by the user. The file CRKDAT contains the analysis control parameters, driver distributions, engineering analysis parameters, and materials information. A complete description of the input data for the CRKDAT data file is given in Section 6.1.3.1.

The results from the PROCRK program are written to *three output files*: CRKRES, IOUTPR, and LOWLIF. CRKRES contains the echo of the information in CRKDAT and the results of the simulation. File IOUTPR contains an echo of the analysis parameters and, if requested, a dump of intermediate calculations. If the program terminates prematurely, an error message will be printed in the IOUTPR file. A list of error messages and possible remedies for the problems is given in Section 6.1.6. LOWLIF contains the first one percent of the lives of the simulated failure distribution.

6.1.3 Description of Input Data Files

Annotated examples of the complete CRKDAT data file format structure for the HEX coil and EXHEX problems are presented in Figures 6.1-1 and 6.1-2, respectively. The data lines of the input files are given in boxes, with a description of each data line located above or adjacent to each box. The specific input parameters of Figures 6.1-1 and 6.1-2 are individually defined in Section 6.1.3.1. Input parameter values given in Figures 6.1-1 and 6.1-2 are not necessarily those used in the application case studies of Section 2.

The input data is read by free format statements from file CRKDAT. Thus, the numbers may be provided sequentially on a line up to 80 characters in length, with each number separated by a blank character or comma. Each number may also be on a separate line in the file. However, it is recommended that the input format suggested in Figures 6.1-1 and 6.1-2 be followed whenever possible.

6.1.3.1 Input File CRKDAT

The required data for the CRKDAT file is divided into the four blocks shown in Figure 6.1-3: analysis parameters, driver information, load and stress, and materials information. The analysis parameters block contains the analysis parameters and the keys to select the program options. The driver information block contains the parameters that define the driver distributions. The number of dynamic loads, the magnitudes of the dynamic loads/stresses, the load file names, the static loads/stresses, and geometry are given in the load and stress block. The materials information block contains the *da/dN* vs. ΔK data, the stress ratio, and the yield strength.

The input parameters are described below by using the following convention: the input variable names are indicated by **BOLD UPPERCASE** letters; the variable types are specified as character [CHR], integer [INT], real [RE], and double precision real [DRE]; the function of the variable is <u>underlined</u> and followed by a description and a list of options, when appropriate; the program and file names are indicated by UPPERCASE letters. A consistent set of units is given in parentheses for specifying dimension, load, stress, and stress intensity factor input parameters. All character strings must be enclosed by 'single quotes'. The user is reminded about the difference between the number "0" and the letter "O" when preparing the input files.

1	Problem type
1	Crack growth model type
675	Random number seed
0	Output dump controller
1	Inner loop size
10000	Outer loop size
1	Growth retardation switch (on)
1	Neuber's rule switch (on)
5	Number of B-lives

Decimal equivalent of percentages for B-lives

0.0001	0.0005	0.001	0.005	0.01

Two Beta distributions on weld offset information

0.06	0.06	0.00	0.00	0.0	0.0
0.00	0.00	0.00	0.00	0.0	0.0
1.00					

Beta distribution on duct inside diameter information

0.1885	0.1915	0.50	0.50	0.5	20.

Beta distribution on wall thickness information

0.0113	0.0157	0.27273	0.27273	0.5	20.

Beta distribution on initial crack aspect ratio information

0.20	1.00	0.50	0.50	0.0	0.0

Beta distribution on initial crack size information

0.005	0.005	0.0	0.0	0.0	0.0

2.00	2.00	0.15	1.00	Narrow-band random load scale factor
2.00	2.00	0.20	1.00	Sinusoidal load scale factor
486.	666.	29.	56.5	Normal distribution on inner wall temperature information
799.	908.	49.5	48.	Normal distribution on outer wall temperature information
3808.	4177.	69.	69.	Normal distribution on internal pressure information

0.80	1.20	Uniform distribution bounds for weld offset accuracy factor
0.50	1.50	Uniform distribution bounds for aerodynamic load scale factor
0.80	1.20	Uniform distribution bounds for aerostatic load scale factor
0.90	1.10	Uniform distribution bounds for aeroloads stress analysis accuracy factor

Figure 6.1-1 Format for File CRKDAT for HEX Coil Problem

0.80	1.20	Uniform distribution bounds for dynamic stress analysis accuracy factor
0.60	1.40	Uniform distribution bounds for Neuber's rule accuracy factor
0.00	0.00	Uniform distribution bounds for threshold stress intensity factor uncertainty
1.00	1.00	Uniform distribution bounds for critical stress intensity factor uncertainty
0.90	1.10	Uniform distribution bounds for stress intensity factor calculation accuracy
-1.38629	0.95166	Uniform distribution bounds for crack growth calculation accuracy factor
3		Number of dynamic loads

Aerostatic load: P, M_x, M_y, M_z, V_y, V_z

0.00	0.00	-0.07214	0.00	0.00	0.00

Dynamic loads: file name, load type, P, M_x, M_y, M_z, V_y, V_z

'NBM3'	1	0.00	0.00	0.00	0.355475	0.00	0.00
'SIN1'	2	0.027374	0.000451	0.001621	0.082116	0.205288	0.005789
'AERO1'	3	0.00	0.00	0.00	0.07179	0.00	0.0

3640	External pressure, p_o
2	Critical duct location
10.	Angular position about the duct circumference, ϕ
2.3	Willenborg retardation model constant
1.0	Reference time history period, T
0.0	Noise filter
20001	Number of points in reference time histories

29000000.	8.8E-06	0.30	E, α, ν

0.615	2.00	The 10 points of the piecewise linear F_k vs. R/t curve
0.693	4.80	
0.753	7.20	
0.813	9.60	
0.873	12.50	
0.933	15.80	
0.993	20.00	
1.029	24.00	
1.053	30.00	
1.053	200.00	

Figure 6.1-1 Format for File CRKDAT for HEX Coil Problem (Cont'd)

6		Number of segments in $\sigma\varepsilon$ vs. ε curve
21.95	0.001	$\sigma_1 \varepsilon_1, \varepsilon_1$
55.77	0.002	$\sigma_2 \varepsilon_2, \varepsilon_2$
144.85	0.005	$\sigma_3 \varepsilon_3, \varepsilon_3$
322.73	0.010	$\sigma_4 \varepsilon_4, \varepsilon_4$
1945.90	0.050	$\sigma_5 \varepsilon_5, \varepsilon_5$
50688.0	0.660	$\sigma_6 \varepsilon_6, \varepsilon_6$

Description of material data

'400 F 316L WELDED FROM RkD'

Materials information: yield strength, critical stress intensity factor, number of data divisions, and regression option

27000 80.0 2 4

Threshold stress intensity factor range model parameters: ΔK_{THo}, C_{α}, d

4.0317 1.070 0.16327

Materials information for first data division: number of points in data division and stress ratio

16 0.90

da/dN vs. ΔK data for division 1

9.183E-10	2.56
1.138E-8	2.69
3.362E-8	2.82
8.473E-8	3.00
4.408E-8	3.33
5.838E-8	3.53
5.679E-8	3.74
7.220E-8	3.95
8.202E-8	4.18
7.440E-8	4.42
9.028E-8	4.67
1.133E-7	4.94
1.533E-7	5.22
1.629E-7	5.51
1.727E-7	5.81
2.321E-7	5.99

Figure 6.1-1 Format for File CRKDAT for HEX Coil Problem (Cont'd)

Materials information for second data division: number of points in data division and stress ratio

18	0.70

da/dN vs. ΔK data for division 2

4.661E-9	3.58
2.469E-8	3.80
1.387E-7	4.49
1.162E-7	4.88
1.631E-7	5.28
1.539E-7	5.74
1.562E-7	6.24
1.839E-7	6.77
2.089E-7	7.35
3.497E-7	7.99
2.949E-7	9.37
3.848E-7	10.15
6.968E-7	11.91
8.980E-7	12.87
1.111E-6	13.89
1.380E-6	15.00
2.790E-6	17.49
3.901E-6	18.17

Figure 6.1-1 Format for File CRKDAT for HEX Coil Problem (Cont'd)

2	Problem type
2	Crack growth model type
675	Random number seed
0	Output dump controller
1	Inner loop size
10000	Outer loop size
1	Growth retardation switch (on)
2	Neuber's rule switch (off)
5	Number of B-lives

Decimal equivalent of percentages for B-lives

0.0001	0.0005	0.001	0.005	0.01

Beta distribution information for plate width

0.027	0.033	0.50	0.50	0.0	0.0

Beta distribution information for initial crack size

0.009	0.011	0.5	0.5	0.0	0.0

2.00	2.00	0.15	1.00	Narrow-band random load scale factor
2.00	2.00	0.20	1.00	Sinusoidal load scale factor

0.90	1.10	Uniform distribution bounds for static stress analysis accuracy factor
0.80	1.20	Uniform distribution bounds for dynamic stress analysis accuracy factor
0.00	0.00	Uniform distribution bounds for threshold stress intensity factor uncertainty
1.00	1.00	Uniform distribution bounds for critical stress intensity factor uncertainty
0.90	1.10	Uniform distribution bounds for stress intensity factor calculation accuracy
-1.38629	0.95166	Uniform distribution bounds for crack growth calculation accuracy factor
-1.50	-2.50	Uniform distribution bounds for Forman equation *m* variation
2		Number of dynamic load sources

Static stresses: σ_x, σ_y, σ_z, σ_{xy}, σ_{xz}, σ_{yz}

0.00	0.00	5000.0	0.00	0.00	0.00

Dynamic stresses: file name, load type, σ_x, σ_y, σ_z, σ_{xy}, σ_{xz}, σ_{yz}

'NBSZ'	1	0.00	0.00	552.34	0.00	0.00	0.00
'SIN2'	2	0.00	0.00	495.86	0.00	0.00	0.00

Figure 6.1-2 Format for File CRKDAT for EXHEX Problem

2.3	Willenborg retardation model constant
1.0	Reference time history period, T
0.0	Noise filter
6001	Number of points in reference time histories

Description of material data

'C10100 COPPER FROM NASA/JSC'

Yield strength, K_c, number of data divisions, and regression option

6100	100.0	1	3

Threshold stress intensity factor range model parameters: ΔK_{THo}, C_o, d

2.2642	-2.6912	-0.55288

Regression constraints: m, p, and q

-2.000	0.00	0.00

Materials information for first data division: number of points in data division and stress ratio

8	0.20

da/dN vs. ΔK data for division 1

5.017E-8	3.037
5.900E-8	3.191
9.798E-8	3.607
1.127E-7	3.649
2.397E-7	4.223
4.069E-7	4.864
5.334E-7	5.473
8.762E-7	6.109

Figure 6.1-2 Format for File CRKDAT for EXHEX Problem (Cont'd)

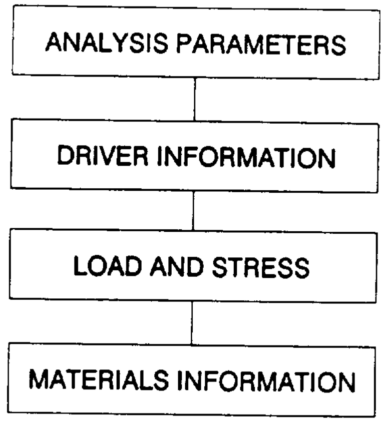


Figure 6.1-3 Data Blocks for Input File CRKDAT

Analysis Parameters Block

KPROB
[INT]

Problem type
PROCRK has the ability to analyze the HEX Coil and the EXHEX. The following integer values control the type of problem.

KPROB = 1 analyze the HEX coil problem

KPROB = 2 analyze EXHEX problem

KGROW
[INT]

Crack growth type
The parameter *m* in the Forman equation may be fixed at the mean value from the regression or may vary between **MVARA** and **MVARB**. Controls the type of *m* parameter variation to be included in the Forman crack growth Equation 2-7.

KGROW = 1 no *m* variation will be included

KGROW = 2 allows Uniform variation in *m*

RAND
[DRE]

Random number seed
Needed by PROCRK's built-in random number generator.

IOUT
[INT]

Output dump controller
PROCRK has the ability to write intermediate calculations to file IOUTPR. The following integer values control the "dump" of PROCRK's calculations.

IOUT	= 0	no intermediate calculation output
IOUT	= 15	driver sampling and driver transformation calculations
IOUT	= 20	crack growth calculations
IOUT	= 25	stress calculations
IOUT	= 30	rainflow cycle counting

NLIFE
[INT]

Inner loop number
Size of the inner loop of the Monte Carlo (MC) simulation. A positive value is required.

NHYPER
[INT]

Outer loop number
Size of the outer loop of the MC simulation. The program requires a positive value.

IRET
[INT]

Crack growth retardation switch
Switch to invoke the Willenborg retardation model in the crack growth calculations. The following integer values control the retardation option.

IRET	= 0	no growth retardation
IRET	= 1	include growth retardation

INEUB
[INT]

Neuber's stress calculation switch
Switch to use the Neuber's rule to calculate an equivalent mean stress. The following integer values control the Neuber's rule option.

INEUB = 0 no Neuber's equivalent mean stress calculation

INEUB = 1 include Neuber's equivalent mean stress calculation

NBLIFE
[INT]

Number of B-lives
The number of B-lives to be provided from the simulated distribution of life. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percentage; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%. **NBLIFE** must be non-negative and cannot exceed 10.

BLFPER(1)	**BLFPER(2) ...**	**BLFPER(NBLIFE)**
[RE]	[RE]	[RE]

B-life percentages
The decimal equivalent of the percentages at which the B-lives are required; e.g., if the B.1 life is desired, then **BLFPER** = 0.001. A total of **NBLIFE** percentages must be provided. The percentage cannot exceed 1% (**BLFPER** ≤ 0.01).

Driver Information Block

WOFFA	**WOFFB**	**WOFFR1**	**WOFFR2**	**WOFFT1**	**WOFFT2**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]
WOFFC	**WOFFD**	**WOFFR3**	**WOFFR4**	**WOFFT3**	**WOFFT4**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]
WOFFE					
[RE]					

Beta distribution on weld offset information
W_{OFF} in Equation 2-3 is the weld offset. It is required for the HEX coil problem (**KPROB** = 1). It may be characterized by two Beta probability distributions. The first two lines are the two Beta distributions, one per line. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The Beta distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for W_{OFF}. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters

describe a Uniform distribution on θ. The third line is the decimal equivalent percentage weight for the first Beta distribution, and it must be between 0.00 and 1.00.

WOFFA	lower bound of the first Beta distribution on W_{OFF}
WOFFB	upper bound of the first Beta distribution on W_{OFF}
WOFFR1	Uniform distribution lower bound of parameter ρ in the first Beta distribution of W_{OFF}
WOFFR2	Uniform distribution upper bound of parameter ρ in the first Beta distribution of W_{OFF}
WOFFT1	Uniform distribution lower bound of parameter θ in the first Beta distribution of W_{OFF}
WOFFT2	Uniform distribution upper bound of parameter θ in the first Beta distribution of W_{OFF}
WOFFC	lower bound of the second Beta distribution on W_{OFF}
WOFFD	upper bound of the second Beta distribution on W_{OFF}
WOFFR3	Uniform distribution lower bound of parameter ρ in the second Beta distribution of W_{OFF}
WOFFR4	Uniform distribution upper bound of parameter ρ in the second Beta distribution of W_{OFF}
WOFFT3	Uniform distribution lower bound of parameter θ in the second Beta distribution of W_{OFF}
WOFFT4	Uniform distribution upper bound of parameter θ in the second Beta distribution of W_{OFF}
WOFFE	decimal equivalent percentage weight occurring in the first Beta distribution of the weld offset, W_{OFF}

INDIAA	**INDIAB**	**INDIR1**	**INDIR2**	**INDIT1**	**INDIT2**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Beta distribution on duct inside diameter information

D_i (in.), the duct inside diameter. It is required for the HEX coil problem (**KPROB** = 1). It is used to calculate R_i in Equation 2-1 and is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the duct inside diameter. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

INDIAA	lower bound of the Beta distribution on D_i

INDIAB	upper bound of the Beta distribution on D_i
INDIR1	Uniform distribution lower bound of parameter ρ in the Beta distribution of D_i
INDIR2	Uniform distribution upper bound of parameter ρ in the Beta distribution of D_i
INDIT1	Uniform distribution lower bound of parameter θ in the Beta distribution of D_i
INDIT2	Uniform distribution upper bound of parameter θ in the Beta distribution of D_i

THICA	**THICB**	**THICR1**	**THICR2**	**THICT1**	**THICT2**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Beta distribution on wall thickness information

t (in.), the duct wall thickness. It is required for the HEX coil problem (**KPROB** = 1). It is used to calculate the area and calculate R_o in Equation 2-1 and is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the wall thickness. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

THICA	lower bound of the Beta distribution on t
THICB	upper bound of the Beta distribution on t
THICR1	Uniform distribution lower bound of parameter ρ in the Beta distribution of t
THICR2	Uniform distribution upper bound of parameter ρ in the Beta distribution of t
THICT1	Uniform distribution lower bound of parameter θ in the Beta distribution of t
THICT2	Uniform distribution upper bound of parameter θ in the Beta distribution of t

AOCA	**AOCB**	**AOCR1**	**AOCR2**	**AOCT1**	**AOCT2**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Beta distribution on initial crack aspect ratio information

a/c, the initial aspect ratio of an elliptic crack. It is required for the HEX coil problem (**KPROB** = 1). It is used to calculate the initial half crack length c_i given the initial

crack depth a_i and is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the aspect ratio. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

AOCA	lower bound of Beta distribution on *a/c*
AOCB	upper bound of Beta distribution on *a/c*
AOCR1	Uniform distribution lower bound of parameter ρ in the Beta distribution of *a/c*
AOCR2	Uniform distribution upper bound of parameter ρ in the Beta distribution of *a/c*
AOCT1	Uniform distribution lower bound of parameter θ in the Beta distribution of *a/c*
AOCT2	Uniform distribution upper bound of parameter θ in the Beta distribution of *a/c*

WITHA	**WITHB**	**WITHR1**	**WITHR2**	**WITHT1**	**WITHT2**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

<u>Beta distribution on plate width information</u>

W (in.), the plate width. It is required for the EXHEX problem (**KPROB** = 2). It is used to calculate the stress intensity factor coefficients and is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the width. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

WITHA	lower bound of the Beta distribution on *W*
WITHB	upper bound of the Beta distribution on *W*
WITHR1	Uniform distribution lower bound of parameter ρ in the Beta distribution of *W*
WITHR2	Uniform distribution upper bound of parameter ρ in the Beta distribution of *W*
WITHT1	Uniform distribution lower bound of parameter θ in the Beta distribution of *W*
WITHT2	Uniform distribution upper bound of parameter θ in the Beta distribution of *W*

AIA	AIB	AIR1	AIR2	AIT1	AIT2
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Beta distribution on initial crack size information

a_i (in.), the initial crack depth for an elliptic crack in the HEX coil problem (**KPROB** = 1) or half the crack length for the EXHEX problem (**KPROB** = 2). It is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the initial crack size. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

AIA lower bound of the Beta distribution on a_i

AIB upper bound of the Beta distribution on a_i

AIR1 Uniform distribution lower bound of parameter ρ in the Beta distribution of a_i

AIR2 Uniform distribution upper bound of parameter ρ in the Beta distribution of a_i

AIT1 Uniform distribution lower bound of parameter θ in the Beta distribution of a_i

AIT2 Uniform distribution upper bound of parameter θ in the Beta distribution of a_i

LAMNA	LAMNB	LAMNC	LAMND
[RE]	[RE]	[RE]	[RE]

Distribution on narrow-band random load scale factor information

This line contains the parameters to define the narrow-band random load scale factor, $\lambda_{D_{RANDOM}}$ in Equation 2-5. See Section 2.1.3.2 in [1] on load scale factors for a detailed description of the parameters k, coefficient of variation C, and strain gage factor d.

LAMNA lower bound of Uniform distribution of k for the narrow-band random load scale factor

LAMNB upper bound of Uniform distribution of k for the narrow-band random load scale factor

LAMNC coefficient of variation C for the narrow-band random load scale factor

LAMND strain gage factor d for the narrow-band random load scale factor

LAMSA **LAMSB** **LAMSC** **LAMSD**
[RE] [RE] [RE] [RE]

Distribution on sinusoidal load scale factor information

This line contains the parameters to define the sinusoidal load scale factor, $\lambda_{D_{SINUSOIDAL}}$ in Equation 2-5. See Section 2.1.3.2 in [1] on load scale factors for a detailed description of the parameters *k*, coefficient of variation *C*, and strain gage factor *d*.

LAMSA	lower bound of Uniform distribution of *k* for the sinusoidal load scale factor
LAMSB	upper bound of Uniform distribution of *k* for the sinusoidal load scale factor
LAMSC	coefficient of variation *C* for the sinusoidal load scale factor
LAMSD	strain gage factor *d* for the sinusoidal load scale factor

TIMUA **TIMUB** **TISIGA** **TISIGB**
[RE] [RE] [RE] [RE]

Normal distribution on inner wall temperature information

T_i (°R), the inner wall temperature. It is required for the HEX coil problem (**KPROB** = 1). It is used to calculate the temperature difference across the wall of the duct, ΔT (°R) in Equation 2-2, and is characterized by a Normal distribution.

TIMUA	Uniform distribution lower bound of parameter μ in the Normal distribution of T_i
TIMUB	Uniform distribution upper bound of parameter μ in the Normal distribution of T_i
TISIGA	Uniform distribution lower bound of parameter σ in the Normal distribution of T_i
TISIGB	Uniform distribution upper bound of parameter σ in the Normal distribution of T_i

TOMUA **TOMUB** **TOSIGA** **TOSIGB**
[RE] [RE] [RE] [RE]

Normal distribution on outer wall temperature information

T_o (°R), the outer wall temperature. It is required for the HEX coil problem (**KPROB** = 1). It is used to calculate the temperature difference across the wall of the duct, ΔT (°R) in Equation 2-2, and is characterized by a Normal distribution.

TOMUA Uniform distribution lower bound of parameter μ in the Normal distribution of T_o

TOMUB Uniform distribution upper bound of parameter μ in the Normal distribution of T_o

TOSIGA Uniform distribution lower bound of parameter σ in the Normal distribution of T_o

TOSIGB Uniform distribution upper bound of parameter σ in the Normal distribution of T_o

PCMUA	**PCMUB**	**PCSIGA**	**PCSIGB**
[RE]	[RE]	[RE]	[RE]

Normal distribution on internal pressure information

p_i (psi) in Equation 2-1. It is required for the HEX coil problem (**KPROB** = 1). This is the inner wall pressure, and it is characterized by a Normal distribution.

PCMUA Uniform distribution lower bound of parameter μ in the Normal distribution of p_i

PCMUB Uniform distribution upper bound of parameter μ in the Normal distribution of p_i

PCSIGA Uniform distribution lower bound of parameter σ in the Normal distribution of p_i

PCSIGB Uniform distribution upper bound of parameter σ in the Normal distribution of p_i

LAMWA	**LAMWB**
[RE]	[RE]

Weld offset stress accuracy factor Uniform distribution information

λ_{OFF} in Equation 2-3. It is required for the HEX coil problem (**KPROB** = 1). This is the weld offset stress concentration accuracy factor, and it is characterized by a Uniform distribution.

LAMWA Uniform distribution lower bound of λ_{OFF}

LAMWB Uniform distribution upper bound of λ_{OFF}

AERDA **AERDB**
[RE] [RE]

Aerodynamic load scale factor distribution information
$\lambda_{D_{AERO}}$ in Equation 2-5. It is required for the HEX coil problem (**KPROB** = 1). This is the aerodynamic load scale factor, and it is characterized by a Uniform distribution.

AERDA	Uniform distribution lower bound of aerodynamic load scale factor
AERDB	Uniform distribution upper bound of aerodynamic load scale factor

AERSA **AERSB**
[RE] [RE]

Aerostatic load scale factor distribution information
$\lambda_{ST_{AERO}}$ in Equation 2-5. It is required for the HEX coil problem (**KPROB** = 1). This is the aerostatic load scale factor, and it is characterized by a Uniform distribution.

AERSA	Uniform distribution lower bound of aerostatic load scale factor
AERSB	Uniform distribution upper bound of aerostatic load scale factor

ASTRA **ASTRB**
[RE] [RE]

Aeroloads stress analysis accuracy factor Uniform distribution information
$\lambda_{AERO_{str}}$ in Equation 2-5. It is required for the HEX coil problem (**KPROB** = 1). This is the aeroloads stress analysis accuracy factor, and it is characterized by a Uniform distribution.

ASTRA	Uniform distribution lower bound of aeroloads stress analysis accuracy factor
ASTRB	Uniform distribution upper bound of aeroloads stress analysis accuracy factor

SSTRA **SSTRB**
[RE] [RE]

Static stress analysis accuracy factor Uniform distribution information
$\lambda_{ST_{str}}$ in Equation 2-5. This is the static stress analysis accuracy factor, and it is characterized by a Uniform distribution. It is required for the EXHEX channel problem (**KPROB** = 2).

SSTRA	Uniform distribution lower bound of static stress analysis accuracy factor
SSTRB	Uniform distribution upper bound of static stress analysis accuracy factor

DSTRA **DSTRB**
[RE] [RE]

Dynamic stress analysis accuracy factor Uniform distribution information

$\lambda_{DYN_{str}}$ in Equation 2-5. This is the dynamic stress analysis accuracy factor, and it is characterized by a Uniform distribution.

DSTRA	Uniform distribution lower bound of dynamic stress analysis accuracy factor
DSTRB	Uniform distribution upper bound of dynamic stress analysis accuracy factor

NEUBA **NEUBB**
[RE] [RE]

Neuber's Rule accuracy factor Uniform distribution information

It is required for the HEX Coil problem (**KPROB** = 1) when **INEUB** = 1. This is the Neuber's Rule accuracy factor, λ_{neu}, and it is characterized by a Uniform distribution. Neuber's Rule is described in Section 2.2.1.4 of [1].

NEUBA	Uniform distribution lower bound of Neuber's Rule accuracy factor
NEUBB	Uniform distribution upper bound of Neuber's Rule accuracy factor

LAMKHA **LAMKHB**
[RE] [RE]

Threshold stress intensity factor uncertainty Uniform distribution information

$\lambda_{K_{th}}$ in Equation 2-8. This is the threshold stress intensity factor range accuracy factor, and it is characterized by a Uniform distribution.

LAMKHA	Uniform distribution lower bound of threshold stress intensity factor range uncertainty
LAMKHB	Uniform distribution upper bound of threshold stress intensity factor range uncertainty

LAMKCA **LAMKCB**
[RE] [RE]

Critical stress intensity factor uncertainty Uniform distribution information
λ_{K_c} in Equation 2-8. This is the critical stress intensity factor uncertainty, and it is characterized by a Uniform distribution.

LAMKCA	Uniform distribution lower bound of critical stress intensity factor uncertainty
LAMKCB	Uniform distribution upper bound of critical stress intensity factor uncertainty

KLAMA **KLAMB**
[RE] [RE]

Stress intensity factor calculation accuracy factor Uniform distribution information
This line contains the Uniform distribution bounds for the stress intensity factor calculation accuracy factor, λ_{sif}.

KLAMA	Uniform distribution lower bound on stress intensity factor calculation accuracy factor
KLAMB	Uniform distribution upper bound on stress intensity factor calculation accuracy factor

LAMGRA **LAMGRB**
[RE] [RE]

Crack growth calculation accuracy factor distribution information
This line contains the Uniform distribution bounds in $\log_e$ space for the crack growth calculation accuracy factor, λ_{gro}, in Equation 2-18.

LAMGRA	lower bound on crack growth calculation accuracy factor
LAMGRB	upper bound on crack growth calculation accuracy factor

MVARA **MVARB**
[RE] [RE]

Forman equation parameter *m* Uniform distribution information
This line contains the Uniform distribution bounds for the Forman equation parameter *m* in Equation 2-7. This is required if **KGROW** = 2.

MVARA	Uniform distribution lower bound on Forman constant *m*
MVARB	Uniform distribution upper bound on Forman constant *m*

Load and Stress Block

The input for loads and stresses for the HEX coil problem (**KPROB** = 1) and EXHEX problem (**KPROB** = 2) are different. For the HEX coil problem the beam-end forces (axial force, moments, and shear forces) from a node in a beam finite element mesh were used. For the EXHEX channel the stress components (σ_x, σ_y, etc.) from a node in a three-dimensional finite element mesh were used.

NLOAD
[INT]

Number of dynamic loads
Total number of dynamic or time-varying loads. **NLOAD** cannot exceed 16.

PSTAT	**TSTAT**	**MSTAT(1)**	**MSTAT(2)**	**VSTAT(1)**	**VSTAT(2)**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Aerostatic loads
This line contains the six beam-end force components due to aerostatic loads. It is required for the HEX coil problem (**KPROB** = 1).[1]

PSTAT	*P* (lbs) in Equation 2-1, the static axial load component
TSTAT	M_x (in.-lbs), the static torsional load component
MSTAT(1)	M_y (in.-lbs) in Equation 2-1, the static moment load component about the y-axis
MSTAT(2)	M_z (in.-lbs) in Equation 2-1, the static moment load component about the z-axis
VSTAT(1)	V_y (lbs), the static shear load component along the y-axis
VSTAT(2)	V_z (lbs), the static shear load component along the z-axis

1 PROCRK does not require M_x, V_y, and V_z. Nevertheless, placeholders for these parameters must be included as the crack growth model uses routines M4L1 and M4L2 without modifications.

LDNAME(I)	TYPE(I)	P(I)	T(I)	M(1,I)	M(2,I)	V(1,I)	V(2,I)
[CHR]	[INT]	[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Dynamic loads

This line contains the dynamic load file names, load types, and the six components of the beam-end force magnitudes. It is required for the HEX coil problem (**KPROB** = 1). A total of **NLOAD** lines must be specified (i.e., the value of **I** goes from 1 to **NLOAD**).[2]

LDNAME(I) File name containing the reference time history for load **I**. The file name cannot be more than six characters long and must be enclosed by single quotes.

TYPE(I) Load-type of load **I**, used to assign the appropriate load scale factor
TYPE(I) = 1 Narrow-band random load
TYPE(I) = 2 Sinusoidal load
TYPE(I) = 3 Aerodynamic load

P(I) P (lbs) in Equation 2-1, the dynamic axial load magnitude for load **I**

T(I) M_x (in.-lbs), the dynamic torsional load magnitude for load **I**

M(1,I) M_y (in.-lbs) in Equation 2-1, the dynamic moment load magnitude about the y-axis for load **I**

M(2,I) M_z (in.-lbs) in Equation 2-1, the dynamic moment load magnitude about the z-axis for load **I**

V(1,I) V_y (lbs), the dynamic shear load magnitude along the y-axis for load **I**

V(2,I) V_z (lbs), the dynamic shear load magnitude along the z-axis for load **I**

SXST	SYST	SZST	SXYST	SXZST	SYZST
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Static stresses

This line contains the six stress components due to static loads. It is required for the EXHEX problem (**KPROB** = 2).

SXST σ_x (psi), due to static loads

SYST σ_y (psi), due to static loads

[2] PROCRK does not require M_x, V_y, and V_z. Nevertheless, placeholders for these parameters must be included as the crack growth model uses routines M4L1 and M4L2 without modifications.

SZST	σ_z (psi), due to static loads
SXYST	σ_{xy} (psi), due to static loads
SXZST	σ_{xz} (psi), due to static loads
SYZST	σ_{yz} (psi), due to static loads

LDNAME(I)	**TYPE(I)**	**SX(I)**	**SY(I)**	**SZ(I)**	**SXY(I)**	**SXZ(I)**	**SYZ(I)**
[CHR]	[INT]	[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Dynamic stresses
This line contains the dynamic load file names, load types, and the six stress component magnitudes. It is required for the EXHEX problem (**KPROB** = 2). A total of **NLOAD** lines must be specified (i.e., the value of **I** goes from 1 to **NLOAD**).

LDNAME(I)	File name containing the reference time history for load source **I**. The file name cannot be more than six characters long and must be enclosed by single quotes.
TYPE(I)	Load-type of load **I**, used to assign the appropriate load scale factor **TYPE(I)** = 1 Narrow-band random load **TYPE(I)** = 2 Sinusoidal load
SX(I)	σ_x (psi), due to dynamic load source **I**
SY(I)	σ_y (psi), due to dynamic load source **I**
SZ(I)	σ_z (psi), due to dynamic load source **I**
SXY(I)	σ_{xy} (psi), due to dynamic load source **I**
SXZ(I)	σ_{xz} (psi), due to dynamic load source **I**
SYZ(I)	σ_{yz} (psi), due to dynamic load source **I**

PCO
[RE]

External pressure
p_o (psi) in Equation 2-1. This is the outer wall pressure. It is required for the HEX coil problem (**KPROB** = 1).

LOCAT
[INT]

Critical location
Critical location of interest on the duct wall. It is required for the HEX coil problem (**KPROB** = 1).

LOCAT = 1 outer wall

LOCAT = 2 inner wall

ANGLE
[RE]

Critical angle
ϕ (degrees) in Equation 2-1. This is the angle measured counterclockwise from the Z-direction to the critical circumferential location of the duct. It is required for the HEX coil problem (**KPROB** = 1).

RSO
[RE]

Willenborg retardation model constant
RSO in Equation 2-13. This is the Willenborg retardation model constant.

PERIOD
[RE]

Period
T (sec) in Equation 2-18. This is the period of the reference time histories, and it is required so that life may be provided in seconds.

TRUNC
[RE]

Noise filter
Value (psi) used to filter out the insignificant cycles in the composite stress-time history during rainflow cycle counting.

NRAN
[INT]

Number of history points
Number of points in the reference time history files for the dynamic loads. **NRAN** cannot exceed 20,000.

EM **COEXP** **NU**
[RE] [RE] [RE]

Materials information
This line contains the elastic modulus, coefficient of thermal expansion, and Poisson's ratio. This line is required for the HEX coil problem (**KPROB** = 1).

EM	E (psi) in Equation 2-2, Young's modulus of elasticity
COEXP	α (/°R) in Equation 2-2, the coefficient of thermal expansion
NU	ν in Equation 2-2, the materials Poisson's ratio

FK(I) **RT(I)**
[RE] [RE]

F_k versus R/t curve
F_k versus R/t points for each segment of the curve are used by Equation 2-3 in the weld offset eccentricity stress concentration calculations. It is required for the HEX coil problem (**KPROB** = 1). A block of 10 segments must be provided (i.e., the value of **I** goes from 1 to 10). Both **FK** and **RT** must be positive and increase with increasing **I** (i.e., **I** = 1 is the lower bound of the first segment, and **I** = 10 is the upper bound of the last segment).

FK(I)	$F_k(R/t)$ value
RT(I)	R/t value

NUMSEG
[INT]

Number of segments
The number of piecewise linear segments in the stress-strain versus strain curve provided. It is required for the HEX coil problem (**KPROB** = 1) when **INEUB** = 1.

SE(J) **E(J)**
[RE] [RE]

Stress-strain versus strain curve
$\sigma\varepsilon$ versus ε points for each segment of the σ vs. ε curve are used in the Neuber's Rule calculations. It is required for the HEX coil problem (**KPROB** = 1) when **INEUB** = 1. A block of **NUMSEG** lines must be provided (i.e., the value of **J** goes

from 1 to **NUMSEG**). Both **SE** and **E** must be positive and increase with increasing **J** as PROCRK assumes that the **J** = 0 point is at the origin.

SE(J) value of the product of stress and strain, $\sigma\varepsilon$, at the upper end of the **J**th segment of the stress-strain versus strain curve

E(J) value of the strain ε at the upper end of the **J**th segment of the stress-strain versus strain curve

Materials Information Block

DESCRP
[CHR]

Description of material data
Name and test environment for the material data. This is a character string no more than 40 characters long, enclosed by single quotes.

FTY	**KC**	**NDIV**	**IREGOP**
[RE]	[RE]	[INT]	[INT]

Materials information
Yield strength, critical stress intensity factor, number of divisions of data, and regression option. The data in each division must have the same stress ratio but data with the same stress ratios may be assigned to different divisions if desired (e.g., from different tests). **NDIV** cannot exceed ten.

FTY yield strength (psi)

KC critical stress intensity factor (ksi$\sqrt{\text{in.}}$)

NDIV number of data divisions for the material data

IREGOP regression option to fit the generalized Forman Equation 2-7[3]
IREGOP = 0 fix p regress for C, n, m, q
IREGOP = 1 fix m, p regress for C, n, q
IREGOP = 2 fix q, p regress for C, n, m
IREGOP = 3 fix m, q, p regress for C, n
IREGOP = 4 regress for C, n, m, q, p

[3] When **KGROW** = 1, the selected value of m will supercede the value obtained from the regression.

DKTHO	CO	DEE
[RE]	[RE]	[RE]

Threshold stress intensity factor range model information
The parameters for the threshold stress intensity factor range vs. stress ratio model given by Equation 2-10.

DKTHO	stress intensity factor range, ΔK_{THo}, at $R = 0$ (ksi$\sqrt{\text{in.}}$)
CO	empirical model constant C_o
DEE	empirical model exponent d

PEE
[RE]

p constraint for **IREGOP** = 0
Parameter *p* in the generalized Forman Equation 2-7. This is required when **IREGOP** = 0.

EMM	PEE
[RE]	[RE]

m, *p* constraint for **IREGOP** = 1
Parameters *m* and *p* in the generalized Forman Equation 2-7. This is required when **IREGOP** = 1.

QUE	PEE
[RE]	[RE]

q, *p* constraint for **IREGOP** = 2
Parameters *q* and *p* in the generalized Forman Equation 2-7. This is required when **IREGOP** = 2.

EMM	QUE	PEE
[RE]	[RE]	[RE]

m, *q*, *p* constraint for **IREGOP** = 3
Parameters *m*, *q*, and *p* in the generalized Forman Equation 2-7. This is required when **IREGOP** = 3.

NP(I) **RDATA(1)**
[INT] [RE]

Information for each crack growth rate data division
Number of points and stress ratio for each data division. This line must be provided for each data division **I**. **NP(I)** for each division cannot exceed two hundred.

NP(I)	number of data points in the division
RDATA(I)	stress ratio for the data in the division

DADN(I,J) **DELK(I,J)**
[RE] [RE]

Crack growth *da/dN* vs. ΔK data
Crack growth rate versus stress intensity factor range data points for each data division. A block of **NP(I)** lines must be specified (i.e., the value of **J** goes from 1 to **NP(I)**). This block must be provided for each data division (i.e., the value of **I** goes from 1 to **NDIV**).

DADN(I,J)	crack growth rate *da/dN* (in./cycle)
DELK(I,J)	stress intensity factor range ΔK (ksi$\sqrt{\text{in.}}$)

6.1.3.2 Reference Time History Files

The data format for the reference time history files is given below. There must be **NLOAD** files with the same names, as specified by **LDNAME(I)** in file CRKDAT. Reference time histories are typically generated by program NBSIN described in Sections 4.5, 6.6, and 7.7 of [1].

STRHIS(I,J)
[RE]

The points of the **I**th reference history
The points of the reference time history specified by **LDNAME(I)**. The data is entered one point per line for **J** = 1, ..., **NRAN**.

6.1.4 Options and Capabilities

PROCRK is a Monte Carlo simulation program which generates a sequence of component lives for a particular failure mode, where life is defined as the accumulated operating time at failure. The simulation has a double-loop structure with **NHYPER** outer loops and **NLIFE** inner loops. The simulation size is dependent on

the failure probability at which a life estimate is desired and the precision desired. For the HEX coil and EXHEX applications, single-loop runs with **NHYPER** = 10,000 and **NLIFE** = 1 were used to characterize component reliability, and single-loop runs with **NHYPER** = 1000 and **NLIFE** = 1 were used for the marginal analysis to assess the importance of drivers.

During a run, it may be desirable to "hold" a driver at a *fixed value*. This may be the nominal or median value of the driver. This is done for drivers with a Beta or a Uniform distribution by merely specifying both the upper and lower bounds to be the desired value. For drivers with a Normal distribution, the standard deviation σ, or coefficient of variation *C*, is set at zero, and the mean, μ, is set at the desired value.

The procedure of holding certain drivers at fixed values while letting the other drivers vary according to their probability distributions may be used for driver variation *sensitivity studies*. That is, the effect on life of driver variation may be evaluated by letting it vary while holding other drivers at fixed values.

A printout of intermediate calculations in various parts of the program may be obtained via the **IOUT** option. This output will be printed in the IOUTPR file. It is recommended that such output not be requested when the simulation size is large since the information will be dumped during every simulation loop.

6.1.5 Code Execution Example

The following example run of the crack growth analysis code PROCRK was carried out with random variation of all drivers for the HPOTP heat exchanger coil small tube outlet (**KPROB** = 1). In this example run, 1000 lives were simulated (**NLIFE** = 1 times **NHYPER** = 1000) with no variation in the Forman constant *m*, **KGROW** = 1. The Willenborg retardation model and the Neuber's rule to calculate the mean stress are switched on (**IRET** = 1, **INEUB** = 1). The B-lives[4] to be provided are B.1, B.5, and B1 (**NBLIFE** = 3, **BLFPER(1)** = 0.001, **BLFPER(2)** = 0.005, **BLFPER(3)** = 0.01). The user may refer to Section 2.2 for additional information on the engineering analysis and to Section 2.3 for the results of the case study for this component.

---

[4] A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.

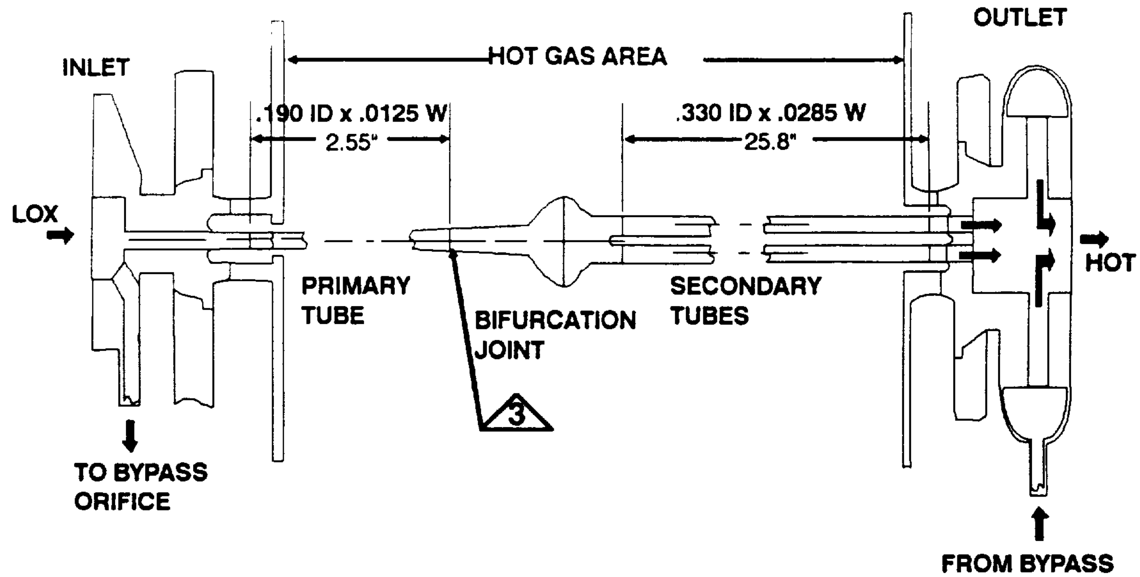


Figure 6.1-4 Detail of the HPOTP Heat Exchanger Coil Small Tube Outlet Near Weld 3

Figure 6.1-4 shows the component in detail and the location of the critical weld, designated as △3. The external pressure **PCO** is 3640 psi. The elastic modulus **EM** is 2.9×10^7, the coefficient of thermal expansion **COEXP** is 8.8×10^{-6}, and Poisson's ratio **NU** is 0.30 for the material.

The drivers for the crack growth failure of weld 3 are as follows:

DRIVER	DISTRIBUTION
Weld Offset	Beta
Inner Diameter	Beta
Wall Thickness	Beta
Initial Crack Aspect Ratio	Beta
Initial Crack Size	Beta
Random & Sine Load Scale Factors	Normal
Flow Conditions (T_i, T_o, p_i)	Normal
Weld Offset Stress Concentration Accuracy	Uniform
Aerodynamic & Static Load Scale Factors	Uniform
Aeroloads & Dynamic Stress Analysis Accuracy Factors	Uniform
Neuber's Rule Accuracy	Uniform
Threshold Stress Intensity Uncertainty Factor	Uniform
Critical Stress Intensity Uncertainty Factor	Uniform
Stress Intensity Factor Calculation Accuracy	Uniform
Growth Calculation Accuracy	Uniform

The rationale for the specification of the driver distributions is given in Section 2.3. The initial crack size was held at 0.005" by fixing the upper and lower bounds of the distribution at **AIA** = **AIB** = 0.005. Also, the weld offset was held at 6% by fixing **WOFFA** = **WOFFB** = 0.06. The threshold stress intensity factor range accuracy was set to **LAMKHA** = **LAMKHB** = 0 resulting in a zero threshold for the crack growth analysis. The critical stress intensity factor accuracy was set to **LAMKCA** = **LAMKCB** = 1.

In addition to the static loads, there were one narrow-band random load, one sinusoidal load, and one aerodynamic load. The three dynamic loads (**NLOAD** = 3) used here are a subset of the loads for this component. The three reference time histories are in the files named NBM3, SIN1, and AERO1, and the contents of these input files are given below. The reference time histories have five points (**NRAN** = 5) and represent 0.00025 seconds (**PERIOD** = 0.00025) of the loading. The reference time histories used for the case studies of the HEX coil small tube outlet given in Section 2.3 consisted of 20,000 points. Shorter histories are used here to permit their inclusion in this example. The critical location is the inner wall (**LOCAT** = 2) at a circumferential position of **ANGLE** = 10°.

The material properties used are for welded 316L stainless steel. The yield strength **FTY** = 27,000 psi, and the critical stress intensity factor **KC** = 80.0 ksi$\sqrt{in}$. The Willenborg retardation model parameter **RSO** = 2.3. The threshold SIF model parameters **DKTHO** = 4.0317, **CO** = 1.070, and **DEE** = 0.16327. Three divisions (**NDIV** = 3) of da/dN vs. ΔK data, which is a subset of the data used in the case study of this component described in Section 2.3, are provided. The first division has 16 data points generated at a stress ratio R = 0.90, the second division has 18 data points at a stress ratio R = 0.70, and the third division has 17 data points at a stress ratio of 0.16. The regression option (**IREGOP** = 4), which derives all the Forman constants C, n, m, p, and q, was used. If further explanation of file CRKDAT is required, refer to Section 6.1.3.1 and Figure 6.1-1.

The echo of the input data is in the output file CRKRES. The simulated B-lives are also given for the component. For instance, the B.1 life is $1.1\text{x}10^5$ seconds. This value is different from the B.1 life obtained in the case study of this component given in Section 2.3 because the number and size of the reference time histories, crack growth rate data points, and the number of simulation trials have been reduced to facilitate the example run. There are only three time histories with just five points each used here, and therefore they do not properly represent the loads for the HEX coil problem. Also, the F_k versus R/t curve is only an example curve.

The IOUTPR file gives an echo of the analysis parameters. The dump parameter **IOUT** is zero; therefore, no other output is in this file. The LOWLIF file contains the lowest one percent of the 1000 simulation lives.

Input File - CRKDAT

```
1
1
675
0
1
1000
1
1
3
0.001  0.005  0.010
0.06     0.06     0.00   0.00    0.0   0.0
0.00     0.00     0.00   0.00    0.0   0.0
1.00
0.1885   0.1915   0.50     0.50     0.5   20.
0.0113   0.0157   0.27273  0.27273  0.5   20.
0.200   1.000   0.50   0.50   0.0   0.0
0.005   0.005   0.00   0.00   0.0   0.0
2.00     2.00     0.15   1.00
2.00     2.00     0.20   1.00
 486.    666.   29.    56.5
 799.    908.   49.5   48.
3808.   4177.   69.    69.
0.80   1.20
0.50   1.50
0.80   1.20
0.90   1.10
0.80   1.20
0.60   1.40
0.00   0.00
1.00   1.00
0.90   1.10
-0.6931   0.557
3
0.00     0.00     -0.07214   0.00      0.00      0.00
'NBM3'  1  0.00      0.00      0.00      0.355475  0.00      0.00
'SIN1'  2  0.027374 0.000451  0.001621  0.082116  0.205288  0.005789
'AERO1' 3  0.00      0.00      0.00      0.07179   0.00      0.00
3640.
2
85.
2.30
0.00025
50.
```

```
5
29000000.   8.8E-06   0.30
0.615     2.00
0.693     4.80
0.753     7.20
0.813     9.60
0.873    12.50
0.933    15.80
0.993    20.00
1.029    24.00
1.053    30.00
1.053   200.00
6
   21.95   0.001
   55.77   0.002
  144.85   0.005
  322.73   0.010
 1945.90   0.050
50688.0    0.660
'400F 316L WELDED, FROM Rkd'
27000  80.0  3  4
4.0317  1.070  0.16327
16  0.90
9.183E-10  2.56
1.138E-8  2.69
3.362E-8  2.82
8.473E-8  3.00
4.408E-8  3.33
5.838E-8  3.53
5.679E-8  3.74
7.220E-8  3.95
8.202E-8  4.18
7.440E-8  4.42
9.028E-8  4.67
1.133E-7  4.94
1.533E-7  5.22
1.629E-7  5.51
1.727E-7  5.81
2.321E-7  5.99
18  0.70
4.661E-9  3.58
2.469E-8  3.80
1.387E-7  4.49
1.162E-7  4.88
1.631E-7  5.28
1.539E-7  5.74
1.562E-7  6.24
1.839E-7  6.77
2.089E-7  7.35
3.497E-7  7.99
```

```
2.949E-7  9.37
3.848E-7  10.15
6.968E-7  11.91
8.980E-7  12.87
1.111E-6  13.89
1.380E-6  15.00
2.790E-6  17.49
3.901E-6  18.17
17  0.16
1.775E-7  9.10
1.969E-7  9.91
2.454E-7  10.79
2.543E-7  11.78
4.050E-7  12.83
5.355E-7  13.96
7.369E-7  15.20
1.058E-6  16.53
2.008E-6  18.00
2.650E-6  19.56
4.238E-6  21.24
5.679E-6  23.11
8.308E-6  25.07
9.687E-6  27.33
1.649E-5  32.96
2.335E-5  38.56
3.304E-5  45.07
```

Input File - NBM3

```
0.862955457680720
0.981515081918201
 1.03346865031769
 1.10476309499562
 1.32048639932450
```

Input File - SIN1

```
-0.976676043502130
-0.931062212127054
-0.862522537797772
-0.772744694860142
-0.663939311885647
```

Input File - AERO1

```
-0.870754448952271
-0.953457959513392
```

```
-0.848940797977113
-0.820263214683910
-0.395816489110529
```

Output File - CRKRES

```
Copyright (C) 1991, California Institute of Technology. U.S. Government
Sponsorship under NASA Contract NAS7-918 is acknowledged.


                         P  R  O  C  R  K

                            INPUT DATA


  DRIVERS                         PARAMETER DISTRIBUTIONS

                                               RHO                  THETA

WELD OFFSET (%)   Be(0.06, 0.06)      U(0.00000, 0.00000)    U( 0.0,   0.0)
                  Be(0.00, 0.00)      U(0.00000, 0.00000)    U( 0.0,   0.0)
                  TEST = 1.00

INNER DIAMETER    Be(0.1885, 0.1915)  U(0.50000, 0.50000)    U( 0.5, 20.0)

WALL THICKNESS    Be(0.0113, 0.0157)  U(0.27273, 0.27273)    U( 0.5, 20.0)

CRACK SHAPE A/C   Be(0.2000, 1.0000)  U(0.50000, 0.50000)    U( 0.0,   0.0)

CRACK SIZE A      Be(0.0050, 0.0050)  U(0.00000, 0.00000)    U( 0.0,   0.0)


LAMBDA RANDOM     k: U(2.00000, 2.00000)
                  COEFFICIENT OF VARIATION: 0.150
                  STRAIN GAGE FACTOR: 1.0000000

LAMBDA SINE       k: U(2.00000, 2.00000)
                  COEFFICIENT OF VARIATION: 0.200
                  STRAIN GAGE FACTOR: 1.0000000


INNER TEMPERATURE    NORMAL:  MU( 486.0,  666.0)   SIGMA( 29.0,  56.5)

OUTER TEMPERATURE    NORMAL:  MU( 799.0,  908.0)   SIGMA( 49.5,  48.0)

INNER PRESSURE       NORMAL:  MU(3808.0, 4177.0)   SIGMA( 69.0,  69.0)
```

WELD OFFSET K FAC	U(0.80000, 1.20000)
DYN AERO LOAD FAC	U(0.50000, 1.50000)
STAT AERO LOAD FAC	U(0.80000, 1.20000)
AERO STR ANAL FAC	U(0.90000, 1.10000)
DYN STR ANAL FAC	U(0.80000, 1.20000)
NEUBERS RULE	U(0.60000, 1.40000)
LAMBDA Kth	U(0.00000, 0.00000)
LAMBDA Kc	U(1.00000, 1.00000)
K CALC FAC	U(0.90000, 1.10000)
GROWTH CALC FAC	U(-0.69310, 0.55700)

LOADS INPUT

	P LOADS	T LOADS	M2 LOADS	M3 LOADS	V2 LOADS	V3 LOADS
STATIC AERO	0.000000	0.000E+00	-.721E-01	0.000E+00	0.000E+00	0.000E+00
NBM3	0.000000	0.000E+00	0.000E+00	0.355E+00	0.000E+00	0.000E+00
SIN1	0.027374	0.451E-03	0.162E-02	0.821E-01	0.205E+00	0.579E-02
AERO1	0.000000	0.000E+00	0.000E+00	0.718E-01	0.000E+00	0.000E+00

MISCELLANEOUS INPUT

EXTERNAL PRESSURE	3640.
ANALYSIS LOCATION	2
ANGLE THETA (DEGREES)	85.0
WILLENBORG OVERLOAD FACTOR	0.23000E+01

STRESS-TIME HISTORY PERIOD	0.00025
STRESS-TIME HISTORY NOISE FILTER	50.0
NUMBER OF TIME-VARYING LOADS	3
NUMBER OF POINTS IN HISTORIES	5

ELASTIC MODULUS	0.290E+08
COEFF OF THERMAL EXPANSION	0.87999997E-05
POISSONS RATIO	0.300

Fk VS. Rt CURVE INPUT

Fk	Rt
0.62	2.00
0.69	4.80
0.75	7.20
0.81	9.60
0.87	12.50
0.93	15.80
0.99	20.00
1.03	24.00
1.05	30.00
1.05	200.00

STRESS-STRAIN CURVE INPUT

MAXIMUM NUMBER OF SEGMENTS 6

STRESS-STRAIN PRODUCT	STRAIN VALUES
21.95	0.00100
55.77	0.00200
144.85	0.00500
322.73	0.01000
1945.90	0.05000
50688.00	0.66000

MATERIAL INPUT

DESCRIPTION: 400F 316L WELDED, FROM Rkd	
YIELD STRENGTH	27000.
CRITICAL S I F	80.
NUMBER OF DIVISIONS	3
REGRESSION OPTION	4

THRESHOLD MODEL DESCRIPTION

DKTHo = 0.40317E+01
Co = 0.10700E+01
d = 0.16327E+00

STRESS RATIO R = 0.90

da/dN	DELK
0.91830E-09	0.25600E+01
0.11380E-07	0.26900E+01
0.33620E-07	0.28200E+01
0.84730E-07	0.30000E+01
0.44080E-07	0.33300E+01
0.58380E-07	0.35300E+01
0.56790E-07	0.37400E+01

0.72200E-07	0.39500E+01
0.82020E-07	0.41800E+01
0.74400E-07	0.44200E+01
0.90280E-07	0.46700E+01
0.11330E-06	0.49400E+01
0.15330E-06	0.52200E+01
0.16290E-06	0.55100E+01
0.17270E-06	0.58100E+01
0.23210E-06	0.59900E+01

STRESS RATIO R = 0.70

da/dN	DELK
0.46610E-08	0.35800E+01
0.24690E-07	0.38000E+01
0.13870E-06	0.44900E+01
0.11620E-06	0.48800E+01
0.16310E-06	0.52800E+01
0.15390E-06	0.57400E+01
0.15620E-06	0.62400E+01
0.18390E-06	0.67700E+01
0.20890E-06	0.73500E+01
0.34970E-06	0.79900E+01
0.29490E-06	0.93700E+01
0.38480E-06	0.10150E+02
0.69680E-06	0.11910E+02
0.89800E-06	0.12870E+02
0.11110E-05	0.13890E+02
0.13800E-05	0.15000E+02
0.27900E-05	0.17490E+02
0.39010E-05	0.18170E+02

STRESS RATIO R = 0.16

da/dN	DELK
0.17750E-06	0.91000E+01
0.19690E-06	0.99100E+01
0.24540E-06	0.10790E+02
0.25430E-06	0.11780E+02
0.40500E-06	0.12830E+02
0.53550E-06	0.13960E+02
0.73690E-06	0.15200E+02
0.10580E-05	0.16530E+02
0.20080E-05	0.18000E+02
0.26500E-05	0.19560E+02
0.42380E-05	0.21240E+02
0.56790E-05	0.23110E+02
0.83080E-05	0.25070E+02

0.96870E-05	0.27330E+02
0.16490E-04	0.32960E+02
0.23350E-04	0.38560E+02
0.33040E-04	0.45070E+02

REGRESSION OUTCOME

c	n	m	p	q
0.56708E-11	0.25314E+01	-0.19413E+01	0.71522E+00	-0.81965E+00

SIMULATION OUTPUT

SHORTEST 1% OF CRACK GROWTH LIVES

LIFE
0.30110E+06
0.37117E+06
0.42265E+06
0.44193E+06
0.44601E+06
0.49042E+06
0.49447E+06
0.49949E+06
0.50079E+06
0.50608E+06

B LIVES:	EMPIRICAL
0.00100	0.30110E+06
0.00500	0.44601E+06
0.01000	0.50608E+06

Output File - IOUTPR

```
PROBLEM TYPE (HEX COIL = 1, EXHEX = 2) =           1
FORMAN EQUATION WITH m (CONST = 1, VARY = 2) =           1
                          RANDOM NUMBER SEED =    675.000000000000
IOUT - OUTPUT CONTROL VARIABLE              =           0
```

```
                   INNER LOOP SIZE  =         1
                   OUTER LOOP SIZE  =      1000
RETARDATION SWITCH  (0 - NO, 1 - YES) =      1
NEUBER SWITCH (0 - NO, 1 - YES)       =      1
```

Output File - LOWLIF

```
 1   0.100000E-02    301098.
 2   0.200000E-02    371174.
 3   0.300000E-02    422653.
 4   0.400000E-02    441930.
 5   0.500000E-02    446013.
 6   0.600000E-02    490422.
 7   0.700000E-02    494468.
 8   0.800000E-02    499490.
 9   0.900000E-02    500786.
10   0.100000E-01    506082.
```

6.1.6 Error Messages and Possible Remedies

The following messages, when applicable, will appear in file IOUTPR. An error message stating that a limit has been exceeded will require that the user increase those limits, as directed, and reviewing or consulting Section 7.1 is desirable. The messages are listed in alphabetical order for the convenience of the user.

ERROR: CANNOT OPEN FILE, 'filename' DOES NOT EXIST

Fatal PROCRK attempted to open the indicated file, however the file did not exist. Check the directory for existence of the file and also check file CRKDAT for correct spelling of the filename.

ERROR: INVALID FORMAN EQUATION SPECIFICATION

Fatal **KGROW** can only have integer values of 1 or 2. Check file CRKDAT for the value used.

ERROR: INVALID LOCATION SPECIFICATION

Fatal **LOCAT** can only have the integer value of 1 or 2. Check file CRKDAT for the value used.

ERROR: INVALID NEUBER'S RULE SWITCH SPECIFICATION

Fatal **INEUB** can only have integer values of 0 or 1. Check file CRKDAT for the value used.

ERROR: INVALID PROBLEM TYPE SPECIFICATION

Fatal **KPROB** can only have integer values of 0 or 1. Check file CRKDAT for the value used.

ERROR: INVALID REGRESSION OPTION SPECIFICATION

Fatal **IREGOP** can only have integer values of 0, 1, 2, 3, or 4. Check file CRKDAT for the value used.

ERROR: INVALID RETARDATION SWITCH SPECIFICATION

Fatal **IRET** can only have integer values of 0 or 1. Check file CRKDAT for the value used.

ERROR: LOAD INCORRECTLY TYPED

Fatal **TYPE(I)** can only have the integer value of 1, 2, or 3. Check file CRKDAT for the value used.

ERROR: NUMBER OF GROWTH RATE DATA POINTS PER DIVISION EXCEEDED

Fatal The materials characterization model cannot accept more than 200 da/dN vs. ΔK points in any data division. It is suggested that the number of data points in each division be recounted. If more than 200 points is desired, the parameter **MAXDAT** must be increased. Refer to Section 7.1 for the routines involved.

ERROR: STRESS-TIME HISTORY TOO LARGE

Fatal No more than 20,000 points is allowed for a reference time history, and an attempt has been made to use a larger history. Check file CRKDAT for a value of **NRAN** larger than 20,000.

K GT Kcr AT A = '**A(1)**'

Warning This is information to the user that the stress intensity factor K exceeded the critical value K_c at crack length **A(1)**, during block growth calculation, for a draw in the simulation.

NO GROWTH AT '**J**'th CRACK LENGTH

Warning This is information to the user that there was no growth at the Jth crack length, during block growth calculation, for a draw in the simulation.

NO GROWTH IN A DIRECTION AT '**J**'th CRACK LENGTH

Warning This is information to the user that there was no growth in the '*a*' direction at the **J**th crack length, during block growth calculation, for a draw in the simulation.

PROGRAM EXECUTION TERMINATED

Fatal This message is produced by routine TRMNAT and follows all other fatal messages.

6.1.7 Summary of Input/Output Files

Input Files

CRKDAT
This file is opened in PROCRK. It contains all parameters for the run options; driver distributions; engineering analysis parameters; and the materials input, including da/dN vs. ΔK data points.

User Specified
These are the reference time history files and are opened in PROCRK. They contain the time histories generated by program NBSIN.

Output Files

CRKRES
This file is opened in PROCRK. It contains the echo of the information contained in CRKDAT, and provides the simulated failure distribution B-life information.[5]

IOUTPR
This file is opened in PROCRK. It contains information on the particular run that is not echoed to CRKRES and the data dump provided when the variable **IOUT** is equal to 15 (Monte Carlo simulation and driver transformation calculations), 20 (crack growth calculations), 25 (stress analysis calculations), or 30 (rainflow cycle counting).

LOWLIF
This file is opened in PROCRK. It contains the lowest one percent of the calculated lives used by the software described in Section 4.2 of [1] to calculate α, β, and θ, the parameters of the Bayesian prior failure distribution.

---

5 A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.

Section 6.2

Low Cycle Fatigue Analysis User's Guide

6.2.1 BLDLCF Program

A user's guide for running the low cycle fatigue (LCF) analysis code BLDLCF is given here. The LCF analysis for the blade is discussed in Section 3, the program description and flowcharts are presented in Section 5.2, and the code structure and listing are provided in Section 7.2.

The BLDLCF program was used to analyze the low cycle fatigue failure of the ATD-HPFTP first stage turbine blade. The output of BLDLCF includes the simulated B-lives and a list of the lowest one percent of lives. The list of lives may be used as input to the regression programs of [1], Section 4.2, to compute the parameters of the Bayesian prior failure distribution. This prior distribution and the success/failure data are used as input to the Bayesian updating program BAYES to obtain a posterior failure distribution. Program Bayes is described in Section 4.3 of [1].

6.2.2 How to Use Program BLDLCF

The program BLDLCF is intended to be run in batch (i.e., background) mode. BLDLCF requires *two input data files*: BLDLCD and RELATD. The materials characterization model portion of the program requires both files for all runs, *even when no related S/N data* is used. The file BLDLCD contains the analysis control parameters, driver distributions, engineering analysis parameters, and specific and exogenous materials information. The file RELATD contains the related materials information. A complete description of the input data for the BLDLCD and RELATD data files is given in Section 6.2.3.

The results from the BLDLCF program are written to *five output files*: BLDLCO, RELATO, DUMP, IOUTPR, and LOWLIF. BLDLCO contains the echo of the information in BLDLCD and the results of the simulation. RELATO contains the echo of the information in RELATD on the related materials data. The results of the materials characterization calculations are primarily given in DUMP. These calculations include point and interval estimates for S/N curve parameters *m* and *C*, posterior credibility ranges for *m*, and an estimate of the median S/N curve. File IOUTPR contains an echo of the analysis parameters and, if requested, a dump of intermediate calculations. If the program terminates prematurely, an error message will be printed in the IOUTPR file. A list of error messages and possible remedies for the problems is given in Section 6.2.6. LOWLIF contains the first one percent of the lives of the simulated failure distribution.

6.2.3 Description of Input Data Files

Annotated examples of the complete data file format structure for BLDLCD and RELATD are presented in Figures 6.2-1 and 6.2-2, respectively. The data lines of the input files are given in boxes, with a description of each data line located adjacent to each box. The specific input parameters of Figures 6.2-1 and 6.2-2 are individually defined in Sections 6.2.3.1 and 6.2.3.2. Input parameter values given in Figures 6.2-1 and 6.2-2 are not necessarily those used in the application case study of Section 3.3.

The input data is read by free format statements from files BLDLCD and RELATD. Thus, the numbers may be provided sequentially on a line up to 80 characters in length, with each number separated by a blank character or comma. Each number may also be on a separate line in the file. However, it is recommended that the input format suggested in Figures 6.2-1 and 6.2-2 be followed whenever possible.

6.2.3.1 Input File BLDLCD

The required data for the BLDLCD file is divided into the four blocks shown in Figure 6.2-3: analysis parameters, driver information, load and geometry, and materials information. The analysis parameters block contains the analysis parameters and the keys to select the program options. The driver information block contains the parameters that define the driver distributions. The parametric sensitivity information, the nominal strains, and reference time history are given in the load and geometry block. The materials information block contains the specific material S/N data including the yield and ultimate strengths, strain ratio, the S/N data points, life region boundaries, and materials characterization model parameter constraints.

The input parameters are described below by using the following convention: the input variable names are indicated by **BOLD UPPERCASE** letters; the variable types are specified as character [CHR], integer [INT], real [RE], and double precision real [DRE]; the function of the variable is underlined and followed by a description and a list of options, when appropriate; the program and file names are indicated by UPPERCASE letters. A consistent set of units is given in parentheses for specifying dimension, load, and strain input parameters. All character strings must be enclosed by 'single quotes'. The user is reminded about the difference between the number "0" and the letter "O" when preparing the input files.

675	Random number seed
0	Value of output dump controller
1	Inner loop size
20000	Outer loop size
50	Symmetry number
2	Type of S/N variation
0	Request for truncated Normal median S/N curve
0	Controls materials process variation
1	Type of materials intrinsic variation
5	Number of B-lives

Decimal equivalent of percentages for B-lives

0.0001	0.0005	0.001	0.005	0.01

Beta distribution on h_{gas} information

676.	2730.	0.50	0.50	0.0	0.0

Beta distribution on T_{gas} information

782.	1982.	0.50	0.50	0.0	0.0

Beta distribution on m information

2730.	2730.	0.50	0.50	0.0	0.0

Beta distribution on λ_G information

0.5	1.5	0.50	0.50	0.0	0.0

Time indices of strain time history defining steady state conditions with stochastic rotor speed given by the included Normal distribution information

5	37592.	507.

0.0	0.020	Normal distribution on e_A information
1640.0	40.67	Normal distribution on T_s information
0.0	0.003	Normal distribution on e_D information
0.0	0.003	Uniform distribution bounds for ε_B
0.96	1.04	Uniform distribution bounds for λ_P
0.80	1.20	Uniform distribution bounds for λ_{MA}
0.975	1.025	Uniform distribution bounds for λ_α
0.70	1.30	Uniform distribution bounds for λ_{TH}
0.00	0.00	Uniform distribution bounds for λ_{dam}
0.00	0.00	Uniform distribution bounds for λ_{TMF}

Figure 6.2-1 Format for File BLDLCD

0.295	38482.	Nominal mechanical strain ε_{Mnom} and corresponding rotor speed ω_o
1.0		Period of reference time history (missions)
0.0		Noise filter (%)
6		Number of points in nominal time history
0.50		Walker exponent w

Coefficients for the start transient response surface function f_A

0.00727362	0.000067442	-0.000059109	-3.52929E-08	1.07611E-08	-2.74419E-08

Coefficients for the shutdown transient response surface functions f_{D1}, f_{D2}, and f_{D3}

-0.132623	0.000227427	-0.000059290	0.0	0.0	4.71714E-08
0.20	950.0				
30523.07	-21846.15				

Nominal time history

225.8	0.0	$\omega(t_1)$, $\varepsilon_{TH}(t_1)$
3025.1	-0.196921	$\omega(t_2)$, $\varepsilon_{TH}(t_2)$
6138.8	0.146025	$\omega(t_3)$, $\varepsilon_{TH}(t_3)$
8309.0	-0.200128	$\omega(t_4)$, $\varepsilon_{TH}(t_4)$
0.0	0.007393	$\omega(t_5)$, $\varepsilon_{TH}(t_5)$

Description of specific material S/N data set

'RT, PWA 1480, 001 DIRECTION'

Specific materials information: yield and ultimate strengths, number of data divisions, and total number of points in data set

1.54	1.57	1	9

Specific materials information for each data division: number of points in data division, strain ratio, and life region

8	-1.0	1

Figure 6.2-1 Format for File BLDLCD (Cont'd)

0.89	6800.	ε_1, N_1
0.89	15000.	ε_2, N_2
0.67	27000.	ε_3, N_3
0.67	43200.	ε_4, N_4
0.56	139300.	ε_5, N_5
0.56	545200.	ε_6, N_6
0.56	147000.	ε_7, N_7
0.39	4344800.	ε_8, N_8
1.57		Strain tensile point
1 0		Number of life regions with and without data
5000.		Life boundary of region 0
1.0E + 36		Life boundary of region 1
0.00		*C* constraint
0 0.000	0.000	Prior information on *m*

- -

0.00	0.00	0.00	Bayesian prior distribution information

- -

0.00	0.00	Materials process variation information

Figure 6.2-1 Format for File BLDLCD (Cont'd)

1 Number of related data sets

'TITANIUM, -423F, 0.14 Fe' Description of related material S/N data set

Related materials information: yield and ultimate strengths, number of data divisions, and total number of points in data set

201700. 215300. 2 10

Related materials information for data division 1: number of points in data division, stress ratio, and life region

4 0.10 1

140000.	38000.	S_1, N_1
130000.	30000.	S_2, N_2
130000.	713000.	S_3, N_3
130000.	310000.	S_4, N_4
6 0.10 2		# points in division 2, stress ratio, region
120000.	72000.	S_5, N_5
110000.	3224000.	S_6, N_6
100000.	910000.	S_7, N_7
100000.	3230000.	S_8, N_8
120000.	665000.	S_9, N_9
110000.	56000.	S_{10}, N_{10}

Figure 6.2-2 Format for File RELATD

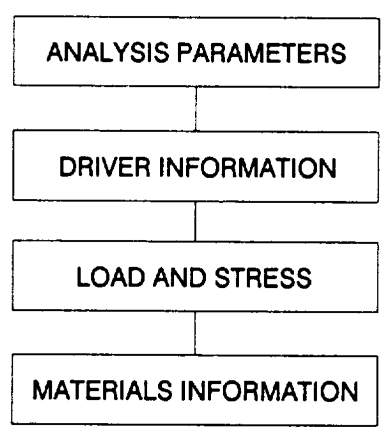


Figure 6.2-3 Data Blocks for Input File

Analysis Parameters Block

RAND
[DRE]

Random number seed
Needed by BLDLCF's built-in random number generator.

IOUT
[INT]

Output dump controller
BLDLCF has the ability to write intermediate calculations to file IOUTPR. The following integer values control the "dump" of BLDLCF's calculations.

IOUT	= 0	no intermediate calculation output
IOUT	= 10	materials characterization model calculations
IOUT	= 15	driver sampling and driver transformation calculations
IOUT	= 20	rainflow cycle counting and damage accumulation

NLIFE
[INT]

Inner loop number
Size of the inner loop of the Monte Carlo (MC) simulation. A positive value is required.

NHYPER
[INT]

Outer loop number
Size of the outer loop of the MC simulation. The program requires a positive value.

NSYM
[INT]

Symmetry number
The number of modeling units in the component. A positive value is required.

VARY
[INT]

Type of S/N variation[6]

Controls the type of stochastic variation to be included in the materials characterization model S/N curve.

VARY	= 0	no variation will be included
VARY	= 1	allows only intrinsic materials variation
VARY	= 2	allows Uniform variation of the materials model shape parameter *m* and intrinsic materials variation
VARY	= 3	allows truncated Normal variation of the materials model shape parameter *m* and intrinsic materials variation
VARY	= 4	allows the variation in the materials model shape parameter *m* to be "bootstrapped"[7]

NMED
[INT]

Request for truncated Normal median S/N curve[8]

If **VARY** = 3, then **NMED** controls the calculation of the empirical median S/N curve.

NMED	= 0	no median curve calculation is required
NMED	= 1	median calculation is required

MPROC
[INT]

Controls materials process variation

Controls the inclusion of materials process variation (heat-to-heat variation). Process variation in materials is discussed in [1], Section 2.1.2.3.

MPROC	= 0	no variation to be included
MPROC	= 1	variation is to be included

VARPHI
[INT]

---

6 A discussion of the possible stochastic specifications of the materials model shape parameter *m* is given in [1], Pages 2-13 through 2-14.

7 This option is only available with program BLDLCF V3.4B1.2.

8 The median S/N curve for the truncated Normal distribution is discussed in [1], Page 2-15.

Type of intrinsic materials variation

Controls the type of intrinsic materials variation to be included in the materials characterization model S/N curve. **VARPHI** is not required if running program BLDLCF V3.4B1.2.

VARPHI	= 1	Weibull intrinsic materials variation will be included
VARPHI	= 2	Lognormal intrinsic materials variation will be included

NBLIFE
[INT]

Number of B-lives

The number of B-lives to be provided from the simulated distribution of life. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percentage; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%. **NBLIFE** must be non-negative and cannot exceed 10.

BLFPER(1)	**BLFPER(2) ...**	**BLFPER(NBLIFE)**
[RE]	[RE]	[RE]

B-life percentages

The decimal equivalent of the percentages at which the B-lives are required; e.g., if the B.1 life is desired, then **BLFPER** = 0.001. A total of **NBLIFE** percentages must be provided. The percentage cannot exceed 50% (**BLFPER** $\leq$ 0.50).

Driver Information Block

HGASA	**HGASB**	**HGASR1**	**HGASR2**	**HGAST1**	**HGAST2**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Beta distribution on h_{gas} information

h_{gas} in Equation 3-2, is the hot gas film coefficient at the start, and it is characterized by a Beta probability distribution. See [1], Section 2.1.3.1 and Equation 2-54, for specifying parameters to define a Beta driver distribution. The Beta distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for h_{gas}. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

HGASA	lower bound of the Beta distribution on h_{gas}
HGASB	upper bound of the Beta distribution on h_{gas}

HGASR1	Uniform distribution lower bound of parameter ρ in the Beta distribution of h_{gas}
HGASR2	Uniform distribution upper bound of parameter ρ in the Beta distribution of h_{gas}
HGAST1	Uniform distribution lower bound of parameter θ in the Beta distribution of h_{gas}
HGAST2	Uniform distribution upper bound of parameter θ in the Beta distribution of h_{gas}

TGASA	**TGASB**	**TGASR1**	**TGASR2**	**TGAST1**	**TGAST2**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Beta distribution on T_{gas} information

T_{gas} in Equation 3-2, is the hot gas temperature at the start, and it is characterized by a Beta probability distribution. See [1], Section 2.1.3.1 and Equation 2-54, for specifying parameters to define a Beta driver distribution. The Beta distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for T_{gas}. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

TGASA	lower bound of the Beta distribution on T_{gas}
TGASB	upper bound of the Beta distribution on T_{gas}
TGASR1	Uniform distribution lower bound of parameter ρ in the Beta distribution of T_{gas}
TGASR2	Uniform distribution upper bound of parameter ρ in the Beta distribution of T_{gas}
TGAST1	Uniform distribution lower bound of parameter θ in the Beta distribution of T_{gas}
TGAST2	Uniform distribution upper bound of parameter θ in the Beta distribution of T_{gas}

SLOPEA	**SLOPEB**	**SLOPR1**	**SLOPR2**	**SLOPT1**	**SLOPT2**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Beta distribution on m information

m (°R/sec) in Equation 3-3, is the deceleration slope at shutdown, and it is characterized by a Beta probability distribution. See [1], Section 2.1.3.1 and Equation 2-54, for specifying parameters to define a Beta driver distribution. The Beta dis-

tribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for m. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

SLOPEA	lower bound of the Beta distribution on m
SLOPEB	upper bound of the Beta distribution on m
SLOPR1	Uniform distribution lower bound of parameter ρ in the Beta distribution of m
SLOPR2	Uniform distribution upper bound of parameter ρ in the Beta distribution of m
SLOPT1	Uniform distribution lower bound of parameter θ in the Beta distribution of m
SLOPT2	Uniform distribution upper bound of parameter θ in the Beta distribution of m

LAMGA	LAMGB	LAMGR1	LAMGR2	LAMGT1	LAMGT2
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

Beta distribution on λ_G information

λ_G in Equation 3-4, is the thermal strain uncertainty factor due to gas temperature variation during the start transient, and it is characterized by a Beta probability distribution. See [1], Section 2.1.3.1 and Equation 2-54, for specifying parameters to define a Beta driver distribution. The Beta distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for λ_G. The next two parameters are the lower and upper bounds for a Uniform distribution on ρ. Similarly, the last two parameters describe a Uniform distribution on θ.

LAMGA	lower bound of Beta distribution on λ_G
LAMGB	upper bound of Beta distribution on λ_G
LAMGR1	Uniform distribution lower bound of parameter ρ in the Beta distribution of λ_G
LAMGR2	Uniform distribution upper bound of parameter ρ in the Beta distribution of λ_G
LAMGT1	Uniform distribution lower bound of parameter θ in the Beta distribution of λ_G
LAMGT2	Uniform distribution upper bound of parameter θ in the Beta distribution of λ_G

TSUBI **SPDMU** **SPDSIG**
[INT] [RE] [RE]

<u>Rotational speed Normal distribution information</u>
The steady state rotational speed is characterized by a Normal(μ, σ^2) distribution. The mean, μ, is equal to the expected operating speed of the turbopump, and the standard deviation, σ, is obtained from the engine performance balance. Both the mean and standard deviation are in rpm.

TSUBI	time index for strain time history for which distribution on steady state speed is valid
SPDMU	mean, μ, of Normally distributed steady state speed
SPDSIG	standard deviation, σ, of Normally distributed steady state speed

FAERRM **FAERRS**
[RE] [RE]

<u>Modeling uncertainty for the start transient Normal distribution information</u>
e_A is the additive modeling uncertainty characterizing the goodness of fit for the start transient response surface given by Equation 3-2. It is characterized by a Normal(μ, σ^2) distribution. The mean, μ, is equal to the expected modeling uncertainty, usually zero, and the standard deviation, σ, is obtained from the curve fitting procedure.

FAERRM	mean, μ, of Normally distributed e_A
FAERRS	standard deviation, σ, of Normally distributed e_A

TSTMU **TSTSIG**
[RE] [RE]

<u>Shutdown transient starting temperature Normal distribution information</u>
T_s (°R) in Equation 3-3. It is the gas temperature at the start of the shutdown transient and is characterized by a Normal(μ, σ^2) distribution. The mean, μ, is equal to the expected T_s, and the standard deviation, σ, is obtained from the engine performance balance.

TSTMU	mean, μ, of Normally distributed T_s
TSTSIG	standard deviation, σ, of Normally distributed T_s

FDERRM **FDERRS**
[RE] [RE]

Modeling uncertainty for the shutdown transient Normal distribution information
e_D is the additive modeling uncertainty characterizing the goodness of fit for the shutdown transient response surface given by Equation 3-3. It is characterized by a Normal(μ, σ^2) distribution. The mean, μ, is equal to the expected modeling uncertainty, usually zero, and the standard deviation, σ, is obtained from the curve fitting procedure.

FDERRM mean, μ, of Normally distributed e_D

FDERRS standard deviation, σ, of Normally distributed e_D

EBENDA **EBENDB**
[RE] [RE]

Bending strain Uniform distribution information
ε_B (%) in Equation 3-1. This is the strain due to gas bending and blade tilt, and it is characterized by a Uniform distribution.

EBENDA Uniform distribution lower bound of ε_B

EBENDB Uniform distribution upper bound of ε_B

LAMPA **LAMPB**
[RE] [RE]

Deviation in blade pull load factor Uniform distribution information
λ_P in Equation 3-5. This is the deviation in blade pull load due to uncertainty in blade mass factor, and it is characterized by a Uniform distribution.

LAMPA variation factor Uniform distribution lower bound of λ_P

LAMPB variation factor Uniform distribution upper bound of λ_P

MANALA **MANALB**
[RE] [RE]

Mechanical strain analysis accuracy factor Uniform distribution information
λ_{MA} in Equation 3-5. This is the mechanical strain analysis accuracy factor, and it is characterized by a Uniform distribution.

MANALA	Uniform distribution lower bound of λ_{MA}
MANALB	Uniform distribution upper bound of λ_{MA}

LAMAA **LAMAB**
[RE] [RE]

Coefficient of thermal expansion variation factor Uniform distribution information
λ_α in Equation 3-4. This is the variation factor for the coefficient of thermal expansion, α, and it is characterized by a Uniform distribution.

LAMAA	Uniform distribution lower bound of λ_α
LAMAB	Uniform distribution upper bound of λ_α

TANALA **TANALB**
[RE] [RE]

Thermal strain analysis accuracy factor Uniform distribution information
λ_{TH} in Equation 3-4. This is the thermal strain analysis accuracy factor, and it is characterized by a Uniform distribution.

TANALA	Uniform distribution lower bound of λ_{TH}
TANALB	Uniform distribution upper bound of λ_{TH}

LAMDAA **LAMDAB**
[RE] [RE]

Damage accumulation model accuracy factor distribution information
This line contains the Uniform distribution bounds in $\log_e$ space for the damage accumulation model accuracy factor, λ_{dam}, in Equation 2-91 of [1]. See [1], Section 2.2.1.4 for a discussion of the damage accumulation calculations.

LAMDAA	lower bound of damage accumulation accuracy factor
LAMDAB	upper bound of damage accumulation accuracy factor

LAMTMA **LAMTMB**
[RE] [RE]

Thermal Mechanical Fatigue model accuracy factor distribution information
This line contains the Uniform distribution bounds in $\log_e$ space for the thermal mechanical fatigue model accuracy factor, λ_{TMF}, in Section 3.2.6.

LAMTMA	lower bound of thermal mechanical fatigue accuracy factor
LAMTMB	upper bound of thermal mechanical fatigue accuracy factor

Load and Geometry Block

EMNOM **NOMSPD**
[RE] [RE]

Nominal mechanical strain and rotor speed
The line contains the nominal mechanical strain, ε_{Mnom} (%), in Equation 3-5, and the nominal or reference rotor speed, ω_o (rpm), corresponding to the nominal mechanical strain value.

PERIOD
[RE]

Period
T (missions) in Equation 2-91 of [1]. This is the period of the nominal strain-time history, and it is required so that life may be provided in missions.

TRUNC
[RE]

Noise Filter
Value (%) used to filter out insignificant cycles in the composite strain-time history during rainflow cycle counting.

NTIME
[INT]

Number of history points
Number of points in the nominal time history. **NTIME** cannot exceed 50.

WEXP
[RE]

<u>Walker exponent</u>
w in Equation 3-8. This is the exponent in the Walker relation used in the equivalent zero mean strain range calculation.

FAA	**FAB**	**FAC**	**FAD**	**FAE**	**FAF**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

<u>Coefficients of the start transient response surface function</u>
The coefficients a, b, c, d, e, and f of the start transient response surface function, Equation 3-2.

$$F_A(T_{gas}, h_{gas}) = a + b\, T_{gas} + c\, h_{gas} + d\, T_{gas}^2 + e\, h_{gas}^2 + f\, T_{gas}\, h_{gas}$$

FD1A	**FD1B**	**FD1C**	**FD1D**	**FD1E**	**FD1F**
[RE]	[RE]	[RE]	[RE]	[RE]	[RE]

FD2A	**FD2B**
[RE]	[RE]

FD3A	**FD3B**
[RE]	[RE]

<u>Coefficients of the shutdown transient response surface functions</u>
The coefficients d_{1A}, d_{1B}, d_{1C}, d_{1D}, d_{1E}, d_{1F}, d_{2A}, d_{2B}, and d_{3A}, d_{3B} of the shutdown transient response surface functions, Equations 3-3, 3-6, and 3-7.

$$f_{D1}(m, T_s) = d_{1A} + d_{1B}\, T_s + d_{1C}\, m + d_{1D}\, T_s^2 + d_{1E}\, m^2 + d_{1F}\, T_s\, m$$

$$t_d = f_{D2}(m, T_s) = d_{2A} + (T_s - d_{2B})/m$$

$$\omega(t_6) = f_{D3}(t_d) = d_{3A} + d_{3B}\, t_d$$

RPM(I) **ETHNOM(I)**
[RE] [RE]

The points of the nominal time history
The points of the nominal time history. The data is entered as rotor speed, thermal strain pairs, one pair per line for **I** = 1, ..., **NTIME**.

Materials Information Block

DESCRP(0)
[CHR]

Description of specific material S/N data set
Name and test environment for the specific material S/N data. This is a character string no more than 40 characters long, enclosed by single quotes.

FTY **FTU** **NDIV** **NPTS(0)**
[RE] [RE] [INT] [INT]

Specific materials information
Yield strength, ultimate strength, number of divisions of data, number of points in S/N data sets. The data may be divided when they are assigned to a different life region or have different strain ratios. If all data has a strain ratio of –1.0, then the yield and ultimate strengths are not required, but zero values must be specified as placeholders. **NPTS(0)** cannot exceed fifty. The next two data sets have to be provided for each data division.

FTY	yield strength corresponding to the specific material data set (%)
FTU	ultimate strength corresponding to the specific material data set (%)
NDIV	number of data divisions for the specific material data set
NPTS(0)	total number of points in the specific material S/N data set

NUM **RATIO** **REG**
[INT] [RE] [INT]

Materials information for each data division of the specific S/N data set
Number of points, strain ratio, and the life region of interest for each data division. This line must be provided for each data division.

NUM	number of S/N data points in the data division

RATIO	strain ratio for the data in the data division
REG	life region number to be assigned to the data in the data division

RAWSTR(I,0) **RAWNF(I,0)**
[RE] [RE]

Specific material S/N data points
Strain versus fatigue life data points for each data division. A block of **NUM** lines must be specified (i.e., the value of **I** goes from 1 to **NUM**). This block must be provided for each data division.

RAWSTR(I,0)	strain value (%)
RAWNF(I,0)	fatigue life value (cycles)

SZERO
[RE]

Tensile point[9]
Strain tensile point, S_o (%). Must be non-negative. A value of zero indicates no tensile point.

NUMREG **NNODAT**
[INT] [INT]

Data regions[10]
Number of life regions that are data-determined and not data-determined. **NUMREG** + **NNODAT** cannot exceed three. **NUMREG** must be 1, 2, or 3, and **NNODAT** must be non-negative, and should be 0 or 1.

NUMREG	number of life regions determined by data
NNODAT	number of life regions (to the right) not determined by data

9 Extension of the S/N curve to the left is discussed in [1], Page 2-17. This option is not available with the "bootstrapping" model.

10 Extension of the S/N curve to the right is discussed in [1], Page 2-17. This option is not available with the "bootstrapping" model.

NBND(L)
[RE]

<u>Life Boundaries</u>[11]

The upper boundaries of the life regions are specified (cycles). The value of **L** goes from **ZROREG** to the total number of regions (equal to **NUMREG** + **NNODAT**). If a non-zero tensile point is specified, then **ZROREG** = 0 else **ZROREG** = 1. The program expects the upper bound of the last life region to be 10^{36}, a proxy for ∞.

CZERO
[RE]

<u>Prior information on coefficient of variation of fatigue strength</u>[12]

Information in the form of a constraint on the coefficient of variation of fatigue strength *C* for the specific material S/N data set. Value must be non-negative and a value of zero indicates **CZERO** is not in use.

MPNT(L)	**MZERO(1,L)**	**MZERO(2,L)**
[INT]	[RE]	[RE]

<u>Prior information on the materials shape parameter *m*</u>[13]

The number of **MZERO** values in each life region, and the lower and upper bound for the range of *m*. The value of **L** goes from 1 to (**NUMREG** + **NNODAT**). If **VARY** = 3 is specified (truncated Normal distribution on *m*), then a prior range of *m* must be specified for each region.

MPNT(L) The number of points, 0, 1, or 2 (no prior on *m*, a point prior on *m*, or a prior over a range of *m*, respectively), in **MZERO( )** for each region.

MZERO(1,L) The lower bound on the range of *m* or the value of the point prior for *m*.

---

[11] Life region boundaries are discussed in [1], Page 2-15.

[12] The implicit constraint on the materials shape parameter provided by prior information on the coefficient of variation of fatigue strength is discussed in [1], Pages 2-12 through 2-13. This option is not available with the "bootstrapping" model.

[13] The explicit constraint on the materials shape parameter provided by prior information on the materials shape parameter is discussed in [1], Page 2-12. This option is not available with the "bootstrapping" model.

MZERO(2,L) The upper bound on the prior range of m. Program requires that the value be zero if a point prior for m is specified.

DELTA(L) **MO(L)** **SIGMA2(L)**
[RE] [RE] [RE]

Information on the Bayesian prior distribution for the truncated Normal distribution[14]

If **VARY** = 3, then the materials model uses the truncated Normal distribution. The truncated Normal distribution requires some prior information on the Normal distribution parameters because a Bayesian analysis is performed. The information is required for each life region. The value of **L** goes from 1 to (**NUMREG** + **NNODAT**).

DELTA(L) The shape parameter, δ, of the Bayesian prior distribution is used to compute the Bayesian posterior distribution parameters. Value must be non-negative. A value of zero indicates a diffuse prior distribution.

MO(L) Location parameter, m_0, of the Bayesian prior distribution of the shape parameter m. Must be positive. Required when **DELTA(L)** is non-zero.

SIGMA2(L) σ^2, the known variance of ln(*fatigue life*), $V(\ln N | \ln S)$. Must be non-negative.

KRATIO **LAMN**
[RE] [RE]

Materials process variation information

If **MPROC** = 1, then specification of **KRATIO** and **LAMN** is required. **KRATIO** is λ_K^*, the ratio *MED* K^*/*MED* K where *MED* K^* is the median value over all heats for the strain (%) at a life of one cycle, and *MED* K is the median value for the specific S/N data for the strain (%) at a life of one cycle. **LAMN** is the ratio of the variance of ln(*life*) conditional on strain over all heats to the intrinsic materials variation for the given S/N data conditional on strain. Process variation in materials is discussed in [1], Section 2.1.2.3.

14 Specification of the Bayesian prior distribution for the truncated Normal case is discussed in [1], Page 2-14.

6.2.3.2 Input File RELATD

The input data for file RELATD, which contains the related materials information,[15] is given below. The data format is similar to that used to specify the S/N data in the specific materials information block in the BLDLCD file.

NSETS
[INT]

<u>Number of related data sets</u>
Number of related material S/N data sets. The following data groups have to be repeated as a block for each data set. The value of **J** varies from 1 to **NSETS**. If there is no related data, then file RELATD will only contain the number "0". **NSETS** cannot exceed five.

DESCRP(J)
[CHR]

<u>Description of related material S/N data set</u>
Name and test environment for related material S/N data set **J**. This is a character string no more than 40 characters long enclosed by single quotes.

FTY	**FTU**	**NDIV**	**NPTS(J)**
[RE]	[RE]	[INT]	[INT]

<u>Related materials information</u>
Yield strength, ultimate strength, number of divisions of data, number of points in S/N data set. The data may be divided when they are assigned to a different life region or have different strain ratios. If all data has a strain ratio of –1.0, then the yield and ultimate strengths are not required, but zero values must be specified as placeholders. **NPTS(J)** cannot exceed fifty. The next two data sets have to be provided for each data division.

FTY	yield strength corresponding to related material data set **J** (%)
FTU	ultimate strength corresponding to related material data set **J** (%)
NDIV	number of data divisions for related material data set **J**
NPTS(J)	total number of points in related material S/N data set **J**

---

[15] Related S/N data is discussed in [1], Page 2-7. This option is not available with the "bootstrapping" model.

NUM **RATIO** **REG**
[INT] [RE] [INT]

Materials information for each data division of the related S/N data set
Number of points, strain ratio, and the life region of interest for each data division. This line must be provided for each data division.

NUM	number of S/N data points in the data division
RATIO	strain ratio for the data in the data division
REG	life region number to be assigned to the data in the data division

RAWSTR(I,J) **RAWNF(I,J)**
[RE] [RE]

Related material S/N data points
Strain versus fatigue life data points for each data division. A block of **NUM** lines must be specified (i.e., the value of **I** goes from 1 to **NUM**). This block must be provided for each data division.

RAWSTR(I,J)	strain value (%)
RAWNF(I,J)	fatigue life value (cycles)

6.2.4 Options and Capabilities

BLDLCF is a Monte Carlo simulation program which generates a sequence of component lives for a particular failure mode, where life is defined as the accumulated operating time at failure. The simulation has a double-loop structure with **NHYPER** outer loops and **NLIFE** inner loops. The simulation size is dependent on the failure probability at which a life estimate is desired and the precision desired. For the blade application, single-loop runs with **NHYPER** = 20,000 and **NLIFE** = 1 were used to characterize component reliability, and single-loop runs with **NHYPER** = 1000 and **NLIFE** = 1 were used for the marginal analysis to assess the importance of drivers.

During a run it may be desirable to "hold" a driver at a *fixed value*. This may be the nominal or median value of the driver. This is done for drivers with a Beta or a Uniform distribution by merely specifying the upper and lower bounds to be the desired value. For drivers with a Normal distribution, the standard deviation, σ, is set at zero, and the mean, μ, is set at the desired value.

The procedure of holding certain drivers at fixed values while letting the other drivers vary according to their probability distributions may be used for driver variation *sensitivity studies*. That is, the effect on life of driver variation may be evaluated by letting it vary while holding other drivers at fixed values. Each driver variation sensitivity was determined in the case studies of this report with the intrinsic variation of the fatigue life of the material included (**VARY** = 1).

A printout of intermediate calculations in various parts of the program may be obtained via the **IOUT** option. This output will be printed in the IOUTPR file. It is recommended that such output not be requested when the simulation size is large since the information will be dumped during every simulation loop. The **NMED** option provides for calculation of an empirical median S/N curve if the truncated Normal distribution or "bootstrapping" is employed.[16] In this case, the median S/N curve is based on the empirical median *m* from all the shape parameters used in the simulation. The **MPROC** option activates the calculations for the process variation feature of the materials characterization model, as discussed in [1], Section 2.1.2.3.

6.2.5 Code Execution Example

The following example run of the LCF analysis code for the ATD-HPFTP first stage turbine blade was carried out with random variation of all drivers. In this example run, 20,000 lives were simulated (**NLIFE** = 1 times **NHYPER** = 20,000) using Uniform shape parameter variation, **VARY** = 2 and **NMED** = 0; Weibull intrinsic materials variation, **VARPHI** = 1; and no materials process variation, **MPROC** = 0. The turbine disk has fifty blades about its circumference, so **NSYM** = 50. The B-lives[17] to be provided are B.1, B.2, B.3, B.4, B.5, B.6, B.7, B.8, B.9, and B1 (**NBLIFE** = 10, **BLFPER(1)** = 0.001, **BLFPER(2)** = 0.002, **BLFPER(3)** = 0.003, **BLFPER(4)** = 0.004, **BLFPER(5)** = 0.005, **BLFPER(6)** = 0.006, **BLFPER(7)** = 0.007, **BLFPER(8)** = 0.008, **BLFPER(9)** = 0.009, **BLFPER(10)** = 0.010). The user may refer to Section 3.2 for additional information on the engineering analysis and to Section 3.3 for the results of the case study for this component.

The drivers for LCF failure of the blade are as follows:

---

16 The truncated Normal distribution for the materials model shape parameter *m* is discussed in [1], Page 2-14.

17 A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.

DRIVER	DISTRIBUTION
h_{gas}	Beta
T_{gas}	Beta
m	Beta
λ_G	Beta
$\omega(t_5)$	Normal
e_A	Normal
T_s	Normal
e_D	Normal
ε_B	Uniform
λ_P	Uniform
λ_{MA}	Uniform
λ_α	Uniform
λ_{TH}	Uniform
λ_{dam}	Uniform
λ_{TMF}	Uniform

The rationale for the specification of the driver distributions is given in Section 3.3.1.

The material is for PWA 1480 tested in the [001] orientation, **DESCRP** = 'RT, PWA 1480, 001 DIRECTION'. The data set includes eight S/N data points, **NUM** = 8, with a strain ratio of −1.0, **RATIO** = −1.0. No strain tensile point is used, **SZERO** = 0, so only one life region upper boundary must be defined, **NBND(0)** = 1.0E36. The number of regions with data, **NUMREG**, is 1, and there are no regions to the right without data, **NNODAT** = 0.[18] The data is in one division, **NDIV** = 1, and the total number of points is eight, **NPTS(0)** = 8. No constraint on the coefficient of variation of fatigue strength is provided, **CZERO** = 0. No explicit range on *m* is included **(MPNT(1)** = **MZERO(1,L)** = **MZERO(2,L)** = 0). No related data is provided. Thus, the RELATD file is empty, except for a single entry to indicate **NSETS** = 0.[19] If further explanation of files BLDLCD and RELATD is required, refer to Sections 6.2.3.1 and 6.2.3.2, and Figures 6.2-1 and 6.2-2, respectively.

---

[18] The nonparametric option is one region only.

[19] The nonparametric option does not use a constraint on the coefficient of variation, an explicit range on *m*, or related data. Nevertheless, placeholders for these parameters must be included because the nonparametric model uses routine RCE without modifications.

The echo of the input data is in the output file BLDLCO. The simulated B-lives are also given for the component. For instance, the B.1 life is 69 missions. The IOUTPR file gives an echo of the analysis parameters. The dump parameter **IOUT** is zero; therefore, no other output is in this file. The LOWLIF file contains the lowest one percent of the 20,000 simulation lives. Finally, the DUMP file contains the results of the materials characterization model information aggregation calculations.[20]

Input File - BLDLCD

```
675
0
1
20000
50
2
0
0
1
10
0.001  0.002  0.003  0.004  0.005  0.006  0.007  0.008  0.009  0.010
 676.   2730.   0.50   0.50   0.00   0.00
 782.   1982.   0.50   0.50   0.00   0.00
2730.   2730.   0.50   0.50   0.00   0.00
  0.5     1.5   0.50   0.50   0.00   0.00
5   37592.   507.
   0.0      0.020
1640.0     40.67
   0.0      0.003
0.00      0.00
0.96      1.04
0.80      1.20
0.975     1.025
0.70      1.30
0.00      0.00
0.00      0.00
0.295
38482.
1.0
0.0
6
0.50
0.00727362    0.000067442    -0.000059109
    -3.52929E-08    1.07611E-08    -2.74419E-08
```

20 The information aggregation calculations are discussed in [1], Pages 2-6 through 2-14.

```
-0.132623    0.000227427   -0.000059290    0.00    0.00    4.71714E-08
0.20       950.0
30523.07  -21846.15
  225.8     0.0
 3025.1   -0.196921
 6138.8    0.146025
 8309.0   -0.200128
    0.0    0.007393
'RT, PWA 1480, 001 DIRECTION'
1.54    1.57    1   8
8   -1.0   1
0.89      6800.
0.89     15000.
0.67     27000.
0.67     43200.
0.56    139300.
0.56    545200.
0.56    147000.
0.39   4344800.
0.00
1  0
1.0E+36
0.00
0   0.000  0.000
```

Input File - RELATO

```
0
```

Output File - BLDLCO

INPUT DATA

DRIVERS		PARAMETER DISTRIBUTIONS	
		RHO	THETA
Hgas	Be(676., 2730.)	U(0.50000, 0.50000)	U(0.0, 0.0)
Tgas (deg R)	Be(782., 1982.)	U(0.50000, 0.50000)	U(0.0, 0.0)
DECEL SLOPE	Be(2730., 2730.)	U(0.50000, 0.50000)	U(0.0, 0.0)

Tgas UNCERT.	Be(0.50, 1.50)	U(0.50000, 0.50000)	U(0.0, 0.0)

	N(MEAN, STD. DEV.)
ROTOR SPEED VARIATION (rpm) AT TIME T5	N(37592.0, 507.0)
Faccel MODELING ERROR	N(0.0, 0.2000E-01)
STARTING DECEL TEMPERATURE (deg R)	N(1640.00, 40.67)
Fdecel MODELING ERROR	N(0.0, 0.3000E-02)
STRAIN DUE TO GAS BENDING (%)	U(0.00000, 0.00000)
LAMBDA BLADE PULL	U(0.96000, 1.04000)
MECHANICAL ANALYSIS FACTOR	U(0.80000, 1.20000)
COEFFICIENT OF THERMAL EXPANSION FACTOR	U(0.97500, 1.02500)
THERMAL ANALYSIS FACTOR	U(0.70000, 1.30000)
DAMAGE MODEL ACCURACY	U(ln 1.00000, ln 1.00000)
TMF MODEL ACCURACY	U(ln 1.00000, ln 1.00000)

OTHER STRAIN HISTORY INPUT

NOMINAL MECHANICAL STRAIN (%)	0.2950
NOMINAL ROTOR SPEED (rpm)	38482.
STRAIN-TIME HISTORY PERIOD (missions)	1.00
STRAIN-TIME HISTORY NOISE FILTER (%)	0.00000
NUMBER OF POINTS IN HISTORIES	6
WALKER EXPONENT	0.50

COEFFICIENTS OF ACCELERATION AND DECELERATION FUNCTIONS

THERMAL STRAIN AT STARTUP (%):

```
   Faccel(Tgas, Hgas) =  0.727362E-02 +  0.674420E-04 * Tgas +
             -0.591090E-04 * Hgas + -0.352929E-07 * Tgas ** 2 +
              0.107611E-07 * Hgas**2 + -0.274419E-07 * Tgas * Hgas
```

THERMAL STRAIN AT SHUTDOWN (%):

```
   Fdecel1(m, Tstart) = -0.132623E+00 +  0.227427E-03 * Tstart +
             -0.592900E-04 * m +  0.000000E+00 * Tstart ** 2 +
              0.000000E+00 * m ** 2 +  0.471714E-07 * Tstart * m
```

TIME AT SHUTDOWN (sec):

```
   Fdecel2(m, Tstart) =  0.200000E+00 + (Tstart -  0.950000E+03 ) / m
```

ROTOR SPEED AT SHUTDOWN (rpm):

```
   Fdecel3(t) =  0.305231E+05 + -0.218462E+05 * t
```

STRAIN HISTORY INFORMATION

ROTOR SPEED rpm	THERMAL STRAIN (%)
225.8	0.000000
3025.1	-0.196921
6138.8	0.146025
8309.0	-0.200128
0.0	0.007393

MATERIAL INPUT

DESCRIPTION: RT, PWA 1480, 001 DIRECTION

YIELD STRENGTH	0.15400E+01
ULTIMATE STRENGTH	0.15700E+01
NUMBER OF POINTS	8

ORIGINAL S/N   STRESS   TRANSFORMED S/N

STRESS	LIFE	RATIO	REGION	STRESS	LIFE
0.89000E+00	6800.	-1.00	1	0.89000E+00	6800.
0.89000E+00	15000.	-1.00	1	0.89000E+00	15000.
0.67000E+00	27000.	-1.00	1	0.67000E+00	27000.
0.67000E+00	43200.	-1.00	1	0.67000E+00	43200.
0.56000E+00	139300.	-1.00	1	0.56000E+00	139300.
0.56000E+00	545200.	-1.00	1	0.56000E+00	545200.
0.56000E+00	147000.	-1.00	1	0.56000E+00	147000.
0.39000E+00	4344800.	-1.00	1	0.39000E+00	4344800.

THERE IS 1 REGION(S) WITH DATA
AND 0 REGION(S) TO THE RIGHT WITHOUT DATA
THE UPPER BOUND(S) OF THE REGION(S) ARE (CYCLES):

0.100E+37

EXOGENOUS INFORMATION

CONSTRAINT ON COEFFICIENT OF VARIATION, C: 0.0000

EXPLICIT CONSTRAINT ON m FOR EACH REGION:

REGION	# OF POINTS	LOWER BOUND	UPPER BOUND
1	0	0.0000	0.0000

WEIBULL VARIATION

B LIVES:	EMPIRICAL
0.00100	0.693627E+02
0.00200	0.104496E+03
0.00300	0.141498E+03
0.00400	0.171753E+03
0.00500	0.203323E+03
0.00600	0.223266E+03
0.00700	0.244718E+03
0.00800	0.266518E+03
0.00900	0.286573E+03
0.01000	0.314683E+03
0.50000	0.900345E+04

Output File - RELATO

NUMBER OF DATA SETS: 0

NOTE: ALL Kt ASSUMED TO BE 1.0

TRANSFORMED DATA

Output File - DUMP

RESULTS OF INFORMATION AGGREGATION CALCULATIONS

95% CONFIDENCE INTERVALS ON C AND m FOR EACH REGION

```
REGION:  1        Io = ( 0.054422790, 0.185977300)
                  Jo = ( 5.152009000, 9.564463000)
```

POINT ESTIMATES OF C AND m FOR EACH REGION

REGION	E(C)	E(m)
1	0.084455910	7.358236

POSTERIOR CREDIBILITY RANGE ON m FOR EACH REGION

REGION	LOWER BOUND	UPPER BOUND
1	5.1520	9.5645

PARAMETER VALUES FOR MEDIAN S/N CURVE

```
NUMBER OF REGIONS:   1      E(BETAo) = 15.7104      E(k) =  1.0909

REGION        m              K            LIFE BOUND        STRESS BOUND

   1       7.35824      0.30172E+01      0.100E+37          0.00000E+00
```

Output File - IOUTPR

```
                         RANDOM NUMBER SEED =    675.000000000000
IOUT (MATCHR = 10, BLDLCF = 15, RAINF3 = 20)=          0
                            INNER LOOP SIZE =          1
                            OUTER LOOP SIZE =      20000
                            SYMMETRY NUMBER =         50
            TYPE OF S/N VARIATION DESIRED
 (0-NONE; 1-INTRINSIC; 2-UNIFORM; 3-NORMAL) =          2
     NORMAL MEDIAN CURVE (0 - NO, 1 - YES) =           0
       MATERIALS PROCESS VARIATION DESIRED
                        (0 - NO, 1 - YES) =            0
       TYPE OF INTRINSIC VARIATION DESIRED
             (1 - WEIBULL; 2 - LOGNORMAL) =            1
```

Output File - LOWLIF

1	0.500000E-04	11.4674
2	0.100000E-03	20.5764
3	0.150000E-03	20.9020
4	0.200000E-03	23.3439
5	0.250000E-03	28.7136
6	0.300000E-03	33.3230
7	0.350000E-03	35.4286
8	0.400000E-03	37.5925
9	0.450000E-03	45.9977
10	0.500000E-03	50.0363
11	0.550000E-03	50.1602
12	0.600000E-03	50.6590
13	0.650000E-03	54.5432
14	0.700000E-03	54.9887
15	0.750000E-03	56.3990
16	0.800000E-03	57.8591
17	0.850000E-03	62.6331
18	0.900000E-03	65.5875
19	0.950000E-03	68.4943
20	0.100000E-02	69.3627
21	0.105000E-02	73.8416
22	0.110000E-02	74.9508
23	0.115000E-02	75.4585
24	0.120000E-02	78.1945
25	0.125000E-02	82.3033

26	0.130000E-02	84.9180
27	0.135000E-02	85.5436
28	0.140000E-02	87.7353
29	0.145000E-02	88.8890
30	0.150000E-02	93.2934
31	0.155000E-02	93.3853
32	0.160000E-02	96.0268
33	0.165000E-02	96.0511
34	0.170000E-02	96.3106
35	0.175000E-02	98.0476
36	0.180000E-02	99.5991
37	0.185000E-02	101.824
38	0.190000E-02	102.286
39	0.195000E-02	103.012
40	0.200000E-02	104.496
41	0.205000E-02	104.946
42	0.210000E-02	106.325
43	0.215000E-02	110.003
44	0.220000E-02	111.212
45	0.225000E-02	111.670
46	0.230000E-02	113.510
47	0.235000E-02	113.610
48	0.240000E-02	114.501
49	0.245000E-02	116.168
50	0.250000E-02	119.642
51	0.255000E-02	121.653
52	0.260000E-02	126.945
53	0.265000E-02	129.652
54	0.270000E-02	132.441
55	0.275000E-02	132.713
56	0.280000E-02	132.853
57	0.285000E-02	134.850
58	0.290000E-02	136.655
59	0.295000E-02	136.710
60	0.300000E-02	141.498
61	0.305000E-02	146.554
62	0.310000E-02	146.987
63	0.315000E-02	147.589
64	0.320000E-02	154.347
65	0.325000E-02	156.143
66	0.330000E-02	158.882
67	0.335000E-02	159.672
68	0.340000E-02	160.197
69	0.345000E-02	161.686
70	0.350000E-02	164.602
71	0.355000E-02	165.648
72	0.360000E-02	165.831
73	0.365000E-02	165.867
74	0.370000E-02	167.298
75	0.375000E-02	167.348

76	0.380000E-02	169.175
77	0.385000E-02	169.208
78	0.390000E-02	169.766
79	0.395000E-02	169.787
80	0.400000E-02	171.753
81	0.405000E-02	175.717
82	0.410000E-02	176.525
83	0.415000E-02	180.021
84	0.420000E-02	180.784
85	0.425000E-02	181.151
86	0.430000E-02	182.652
87	0.435000E-02	182.757
88	0.440000E-02	183.970
89	0.445000E-02	184.185
90	0.450000E-02	185.089
91	0.455000E-02	186.951
92	0.460000E-02	187.950
93	0.465000E-02	188.068
94	0.470000E-02	191.095
95	0.475000E-02	193.211
96	0.480000E-02	199.758
97	0.485000E-02	200.120
98	0.490000E-02	200.299
99	0.495000E-02	200.820
100	0.500000E-02	203.323
101	0.505000E-02	204.715
102	0.510000E-02	206.620
103	0.515000E-02	208.139
104	0.520000E-02	208.957
105	0.525000E-02	209.029
106	0.530000E-02	209.388
107	0.535000E-02	209.562
108	0.540000E-02	211.436
109	0.545000E-02	212.186
110	0.550000E-02	213.019
111	0.555000E-02	213.384
112	0.560000E-02	215.517
113	0.565000E-02	216.541
114	0.570000E-02	217.368
115	0.575000E-02	219.029
116	0.580000E-02	219.229
117	0.585000E-02	220.573
118	0.590000E-02	221.352
119	0.595000E-02	223.254
120	0.600000E-02	223.266
121	0.605000E-02	224.214
122	0.610000E-02	224.821
123	0.615000E-02	224.941
124	0.620000E-02	225.630
125	0.625000E-02	230.432

126	0.630000E-02	230.894
127	0.635000E-02	232.867
128	0.640000E-02	233.193
129	0.645000E-02	235.233
130	0.650000E-02	235.455
131	0.655000E-02	235.684
132	0.660000E-02	237.135
133	0.665000E-02	239.252
134	0.670000E-02	240.345
135	0.675000E-02	241.842
136	0.680000E-02	242.162
137	0.685000E-02	242.815
138	0.690000E-02	244.131
139	0.695000E-02	244.176
140	0.700000E-02	244.718
141	0.705000E-02	245.149
142	0.710000E-02	248.546
143	0.715000E-02	251.099
144	0.720000E-02	251.607
145	0.725000E-02	251.614
146	0.730000E-02	252.298
147	0.735000E-02	253.937
148	0.740000E-02	255.248
149	0.745000E-02	259.308
150	0.750000E-02	259.677
151	0.755000E-02	260.639
152	0.760000E-02	261.692
153	0.765000E-02	262.321
154	0.770000E-02	263.077
155	0.775000E-02	263.105
156	0.780000E-02	263.857
157	0.785000E-02	265.718
158	0.790000E-02	265.802
159	0.795000E-02	266.451
160	0.800000E-02	266.518
161	0.805000E-02	266.648
162	0.810000E-02	268.302
163	0.815000E-02	268.492
164	0.820000E-02	268.948
165	0.825000E-02	268.991
166	0.830000E-02	269.684
167	0.835000E-02	272.396
168	0.840000E-02	272.490
169	0.845000E-02	273.289
170	0.850000E-02	273.440
171	0.855000E-02	273.690
172	0.860000E-02	275.113
173	0.865000E-02	277.709
174	0.870000E-02	278.107
175	0.875000E-02	279.670

176	0.880000E-02	283.247
177	0.885000E-02	283.595
178	0.890000E-02	284.003
179	0.895000E-02	285.168
180	0.900000E-02	286.573
181	0.905000E-02	289.375
182	0.910000E-02	294.471
183	0.915000E-02	294.922
184	0.920000E-02	296.875
185	0.925000E-02	297.684
186	0.930000E-02	299.328
187	0.935000E-02	299.942
188	0.940000E-02	301.087
189	0.945000E-02	303.201
190	0.950000E-02	303.504
191	0.955000E-02	304.032
192	0.960000E-02	305.626
193	0.965000E-02	306.731
194	0.970000E-02	308.299
195	0.975000E-02	308.348
196	0.980000E-02	309.762
197	0.985000E-02	312.676
198	0.990000E-02	314.043
199	0.995000E-02	314.101
200	0.100000E-01	314.683

6.2.6 Error Messages and Possible Remedies

The following messages, when applicable, will appear in file IOUTPR. These messages are primarily generated by the materials characterization model (MATCHR) portion of BLDLCF. An error message stating that a limit has been exceeded will require that the user increase those limits, as directed, and reviewing or consulting [1], Section 7.3.1.3, is desirable. The messages are listed in alphabetical order for the convenience of the user.

ERROR: BAD VALUE FOR DELTA OR VALUE OF MO INCONSISTENT WITH DELTA IN REGION 'L'

Fatal This error can occur during the use of the truncated Normal variation option of the materials characterization model for two reasons. First, the value of δ may be negative. Second, a value of δ was specified, but the value of m_o is not positive. Check file BLDLCD.

ERROR: Co TOO LOW

Fatal The constraint, C_o, imposed on the coefficient of variation of fatigue strength is inconsistent with the observed S/N data.

ERROR: EXCEEDED LIMIT ON DEGREES OF FREEDOM IN CHI-SQUARE TABLE, IN REGION '**L**'

Fatal As implemented, the credibility interval calculations can handle no more than 150 degrees of freedom, and the amount of data in the region indicated requires more. The χ^2 tables of routine INTRVL must be increased. See [1], Sections 4.1.3.6 and 7.3.1.3, for more information.

ERROR: EXCEEDED LIMIT ON NUMBER OF REGIONS

Fatal The materials characterization model can handle no more than 3 life regions. Check file BLDLCD because the sum of the number of regions with data and the number of regions without data is greater than 3.

ERROR: INVALID RESPONSE TO NORMAL MEDIAN CURVE QUESTION

Fatal **NMED** can only have the integer value 0 or 1. Check file IOUTPR for the value used.

ERROR: INVALID TYPE OF MATERIALS PROCESS VARIATION DESIRED

Fatal **MPROC** can only have the integer value 0 or 1. Check file IOUTPR for the value used.

ERROR: INVALID TYPE OF S/N VARIATION DESIRED

Fatal **VARY** can only have the integer value 0, 1, 2, or 3.[21] Check file IOUTPR for the value used.

ERROR: INVALID VALUE FOR RATIO: '**RATIO**'

Fatal An invalid value for the strain ratio has been declared for the specific material data set. Only values between −1.0 and +1.0 inclusive, are possible. Check file BLDLCD.

ERROR: INVALID VALUE OF RATIO: '**RATIO**'

Fatal An invalid value for the strain ratio has been declared for a related material data set. Only values between −1.0 and +1.0 inclusive, are possible. Check file RELATD.

ERROR: NO INTERSECTION BETWEEN Jo AND Mc
ERROR: NO INTERSECTION BETWEEN Jo AND Mo

---

[21] **VARY** can also have the integer value of 4 if program BLDLCF V3.4B1.2 is being used.

ERROR: NO INTERSECTION BETWEEN Jo, Mo, AND Mc
ERROR: NO INTERSECTION BETWEEN Mo AND Mc

Fatal These errors indicate that the specified C constraint and/or prior credibility range on m do not agree with each other and/or the observed S/N data.

ERROR: NORMAL VARIATION REQUIRES A PRIOR RANGE ON M

Fatal The truncated Normal variation option of the materials characterization model requires a prior range on m. The number of points for the prior range on m has been incorrectly specified. Check file BLDLCD to verify that the number of points indicated for each range has an integer value of 1 or 2.

ERROR: NUMBER OF POINTS PER DIVISION INCORRECTLY SPECIFIED IN SET 'J'

Fatal The materials characterization model has been given conflicting information about the number of points in one of the related S/N data sets. Check file RELATD for each related data set to compare the total number of points declared with the sum of the numbers of points in each data division.

ERROR: NUMBER OF POINTS PER DIVISION INCORRECTLY SPECIFIED IN SPECIFIC DATA SET

Fatal The materials characterization model has been given conflicting information about the number of points in the specific S/N data set. Check file BLDLCD, since the total number of points in the specific data set declared and the sum of the numbers of points in each data division do not agree.

ERROR: OVERALL PRIOR RANGE INCORRECTLY SPECIFIED IN REGION WITHOUT DATA

Fatal The prior credibility range on m in one of the regions without data has been incorrectly specified. Check file BLDLCD to verify that either more regions without data have been indicated than intended or that the number of points in the prior on m in a region without data has been incorrectly specified. Only the integer value 0, 1, or 2 is acceptable.

ERROR: OVER LIMIT ON NUMBER OF POINTS IN SET 'J'

Fatal The materials characterization model cannot accept more than 50 S/N points in any related material data set. Check file RELATD for the total number of points in each related data set declared, or there may be more than 50 S/N points with an incorrect total declaration. It is suggested that the number of S/N data points in each related set be recounted. If more

than 50 points are desired, the parameter **MAXDAT** must be increased. Refer to [1], Section 7.3.1.3, for the routines involved.

ERROR: OVER LIMIT ON NUMBER OF RELATED DATA SETS

Fatal The materials characterization model allows up to 5 related data sets. Check file RELATD to determine if more than 5 related data sets were specified. The parameter **MAXSET** must be increased. Refer to [1], Section 7.3.1.3, for the routines involved.

ERROR: OVER NUMBER OF POINTS LIMIT IN SPECIFIC MATERIAL

Fatal The materials characterization model cannot accept more than 50 S/N points in the specific material data set. Check file BLDLCD for the total number of points in the specific data set declared, or there may be more than 50 S/N points with an incorrect total declaration. If more than 50 points are desired, the parameter **MAXDAT** must be increased. Refer to [1], Section 7.3.1.3, for the routines involved.

ERROR: OVER REGION LIMIT IN RELATED MATERIAL '**J**'

Fatal No more than 3 life regions are allowed, and an attempt has been made to place some S/N data in a region number greater than 3. Check file RELATD for an invalid region number immediately following the strain ratio value in the data set indicated.

ERROR: OVER REGION LIMIT IN SPECIFIC DATA SET

Fatal No more than 3 life regions are allowed, and an attempt has been made to place some S/N data in a region number greater than 3. Check file BLDLCD for an invalid region number immediately following the strain ratio value.

ERROR: POSTERIOR INTERVAL IN REGION '**L**' IS INCONSISTENT WITH POINT POSTERIOR IN REGION '**L**–1'

Fatal Check file DUMP to verify that the point posterior value of *m* in region '**L**–1' is greater than the upper bound of the posterior credibility range in region '**L**'. This error indicates a violation of the concavity assumption.

ERROR: POSTERIOR INTERVAL IN REGION '**L**' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN REGION '**L**–1'

Fatal Check file DUMP to verify that the lower bound of the posterior credibility range of *m* in region '**L**–1' is greater than the upper bound of the posterior credibility range of *m* in region '**L**'. The data should be checked for consistency.

ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN '**L**'

Fatal The number of points for the specified prior range of *m* in the indicated region has been incorrectly provided. Check file BLDLCD to verify that the number of points indicated for each range has an integer value of 0, 1, or 2.

ERROR: STRAIN-TIME HISTORY TOO LARGE

Fatal No more than 50 points is allowed for the nominal time history and an attempt has been made to use a larger history. Check file BLDLCD for a value of **NTIME** larger than 50.

ERROR: SXY > = 0 IN REGION '**L**'

Fatal During the linear regression calculations, for the region indicated, the resulting value of the sample covariance S_{xy} was found to be non-negative. This suggests that the data is specified erroneously or is inadequate for analysis, since life increasing with increasing strain contradicts the true fatigue behavior of materials.

ERROR: TOO FEW POINTS FOR REGRESSION IN REGION '**L**'

Fatal The materials characterization model does not have the required minimum number of points in the region indicated to perform a linear regression. If there are no related data sets, then there must be at least 3 points in each region. If there are *N* related data sets, then the total number of points in each region (specific and related combined) must be at least $N + 3$.

IMPOSSIBLE M RANGE IN REGION '**L**'

Fatal Concavity constraints during the random *m* selection have required an impossible range on *m* for the region indicated. Take note of all input parameters for this run, and consult [1], Sections 4.1.5.1, 4.1.5.2, and 7.3, to aid in identification of the cause of this error.

NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ON m IN REGION '**L**'

Warning This means that the estimate of *m* based on the S/N data only, in the region indicated, is outside the range indicated by the specified constraints on *m* and *C*.

PROCESS EXECUTION TERMINATED

Fatal This message is produced by routine TRMNAT and follows all other fatal messages.

6.2.7 Summary of Input/Output Files

Input Files

BLDLCD
This file is opened in BLDLCF. BLDLCD has the following elements: parameters for the run options; driver distributions; values for nominal strains and their associated parametric sensitivity coefficients; and the specific and exogenous materials input, including yield and ultimate strengths (%), strain ratio, S/N data points, life (cycles) boundaries, region information, coefficient of variation constraint, *C*, and prior ranges on the materials shape parameter *m* for each region.

RELATD
This file is opened in subroutine INFAGG. It contains the related material data input, including yield and ultimate strengths (%), strain ratio, S/N data points, and region information.

Output Files

BLDLCO
This file is opened in BLDLCF. It contains the echo of the information contained in BLDLCD and provides the simulated failure distribution B-life information.[22]

RELATO
This file is opened in subroutine INFAGG. It contains the echo of the information contained in RELATD.

DUMP
This file is opened in BLDLCF. It contains the results of the information aggregation portion of the materials model calculations, such as I_o and J_o; the point estimates of *m* and *C*; posterior credibility ranges for *m*; and a list of the estimated values for all S/N curve parameters. See [1], Section 4.1.

IOUTPR
This file is opened in BLDLCF. It contains information on the particular run that is not echoed to BLDLCO and the data dump provided when the variable **IOUT** is equal to 10 (materials characterization calculations), 15 (Monte Carlo simulation

[22] A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.

and driver transformation calculations), or 20 (cycle counting and damage accumulation).

LOWLIF

This file is opened in BLDLCF. It contains the first one percent of the calculated lives used by the software described in [1], Section 4.2, to calculate α, β, and θ, the parameters of the Bayesian prior failure distribution.

Reference

[1] Moore, N., et al., An Improved Approach for Flight Readiness Certification – Methodology for Failure Risk Assessment and Application Examples, JPL Publication 92-15, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, June 1, 1992.

7.0 Structure and Listing of Programs

Section 7.1

Crack Growth Analysis Software PROCRK

The program tree structures, list of subprograms, descriptions of the key variables, and the FORTRAN source listing for the crack growth analysis code PROCRK are given here. The pertinent crack growth methodology is given in Section 2.2. The overall description of the program and the flowcharts are given in Section 5.1. The user's guide for running PROCRK is given in Section 6.1.

7.1.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for PROCRK is given in Figure 7.1-1. The program, subprogram, and file names are indicated by UPPERCASE letters.

7.1.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7.1-1. The section numbers where the subprograms are described by means of flowcharts are given next to the names.

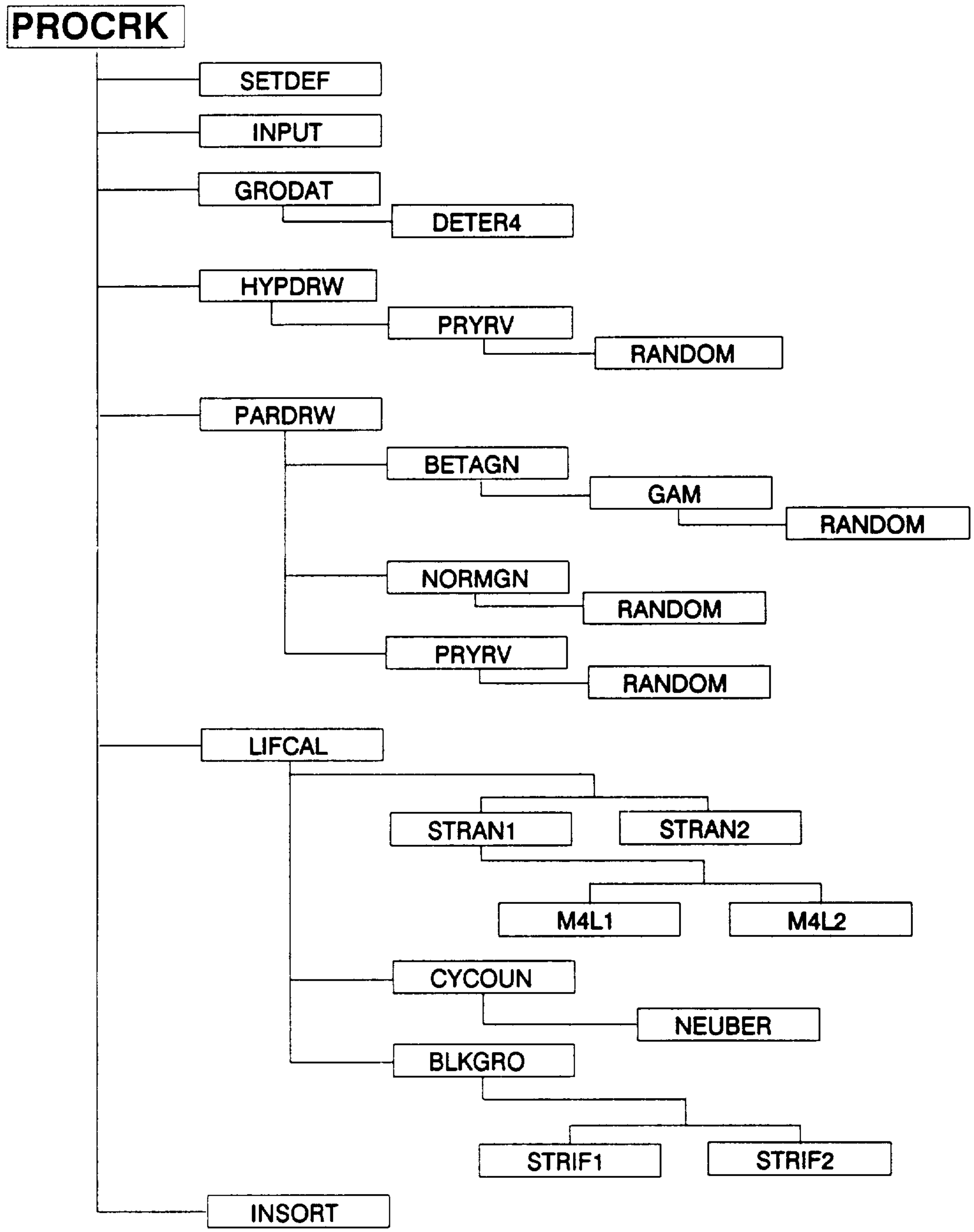


Figure 7.1-1 Tree Structure for Program PROCRK

Table 7.1-1 List of Subprograms For Program PROCRK (Footnotes are at the end of the table)

NAME	SECTION	PURPOSE
BETAGN[1]	4.4.5*	Generates Beta(a, b, ρ, θ) random variates.
BLKGRO	5.1.2.10	Calculates the crack growth rate per load block.
CYCOUN	5.1.2.9	Calculates the number of cycles by rainflow counting, creates a stress vs. cycles table, and determines the equivalent mean stress.
DETER4	5.1.2.4	Calculates the determinant of a 4x4 matrix.
GAM	4.4.4*	Generates Gamma(α, 1) random variates.
GRODAT	5.1.2.4	Reads material properties and performs regression on crack growth data.
HYPDRW	5.1.2.5	Performs hyperparameter draws in the outer loop.
INPUT	5.1.2.3	Reads the data from file CRKDAT and echoes the data to file CRKRES.
INSORT	5.B*	Performs an insertion sort for the lowest one percent of the lives calculated.
LIFCAL	5.1.2.7	Calculates the crack growth life.
M4L1	5.1.3.3*	Performs the driver transformation, for location 1, the exterior surface of the duct.
M4L2	5.1.3.3*	Performs the driver transformation, for location 2, the interior surface of the duct.
NEUBER	5.1.3.6*	Calculates the equivalent mean stress from the maximum stress based on Neuber's rule. See Section 2.2.1.4 of [1].
NORMGN[2]	4.4.3*	Generates Normal(μ, σ^2) random variates.
PARDRW	5.1.2.6	Performs the random life driver parameter draws in the inner loop.
PROCRK	5.1.2.1	The main routine that controls the logical flow of the probabilistic crack growth analysis.
PRYRV[3]	7.6.6*	Generates the Uniform(a, b) and Uniform(c, d) pair of independent random variates.
RANDOM[3]	4.4.2*	Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.
SETDEF	5.1.2.2	Initializes arrays and variables and sets them to default values.
STRAN1	5.1.2.8	Derives the composite principal stress history for the HEX coil.
STRAN2	5.1.2.8	Derives the composite principal stress history for the EXHEX.
STRIF1	5.1.2.11	Calculates the stress intensity factor coefficients for the HEX coil crack configuration.

Table 7.1-1 List of Subprograms For Program PROCRK
(Footnotes are at the end of the table)

NAME	SECTION	PURPOSE
STRIF2	5.1.2.11	Calculates the stress intensity factor coefficients for the EXHEX crack configuration.
TRMNAT	4.1.11*	Performs premature program termination, when required.

---

* See [1].

1 The Beta distribution is discussed in [1], Page 2-25.

2 The Normal distribution is discussed in [1], Page 2-23.

3 The Uniform distribution is discussed in [1], Page 2-23.

7.1.3 Description of Variables

A list of variables used in crack growth analysis code, PROCRK, is given in Table 7.1-2. The variable names are indicated by **BOLD UPPERCASE** letters; the variable "type" can be interpreted as follows: CH6 is a character variable, six characters long; INT is a standard integer variable; LOG is a standard logical variable; RE is a standard real variable; and DRE is a double precision variable. The various array dimensions are defined by using the following parameters: **MAXBLF**, **MAXDAT**, **MAXDIV**, **MAXLD**, **MAXLIF**, **MAXM**, and **MAXSEG**.

Table 7.1-2 List of Variables For Program PROCRK
(Footnotes are at the end of the table)

VARIABLE NAME	TYPE	DESCRIPTION
AERD	RE	λD_{AERO} in Equation 2-5, the randomly selected load scale factor for the AERoDynamic load components.
AERDA	RE	Uniform distribution lower bound of the aerodynamic load scale factor.
AERDB	RE	Uniform distribution upper bound of the aerodynamic load scale factor.
AERS	RE	λST_{AERO} in Equation 2-5, the randomly selected load scale factor for the AERoStatic load components.
AERSA	RE	Uniform distribution lower bound of the aerostatic load scale factor.
AERSB	RE	Uniform distribution upper bound of the aerostatic load scale factor.
AI	RE	a_i (in.), randomly selected initial crack dimension.
AIA	RE	Lower bound of the Beta distribution on a_i.
AIB	RE	Upper bound of the Beta distribution on a_i.
AIR	RE	Randomly selected Beta distribution location parameter ρ for a_i.
AIR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution for a_i.
AIR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution for a_i.
AIT	RE	Randomly selected Beta distribution location parameter θ for a_i.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
AIT1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution for a_i.
AIT2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution for a_i.
ANGLE	RE	ϕ (rad) in Equation 2-1, the angle measured counterclockwise from Z-direction to the critical circumferential location.
AOC	RE	a/c, the randomly selected initial crack aspect ratio.
AOCA	RE	Lower bound of the Beta distribution on a/c.
AOCB	RE	Upper bound of the Beta distribution on a/c.
AOCR	RE	Randomly selected Beta distribution location parameter ρ for a/c.
AOCR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution for a/c.
AOCR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution for a/c.
AOCT	RE	Randomly selected Beta distribution location parameter θ for a/c.
AOCT1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution for a/c.
AOCT2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution for a/c.
ASTR	RE	$\lambda_{AERO_{str}}$ in Equation 2-5, the randomly selected aerodynamic stress analysis accuracy factor.
ASTRA	RE	Uniform distribution lower bound of the aerodynamic stress analysis accuracy factor.
ASTRB	RE	Uniform distribution upper bound of the aerodynamic stress analysis accuracy factor.
BLFPER(MAXBLF)	RE	1-D array containing user-specified B-lives which are obtained from the simulated failure distribution. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent: e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.
BLFPOS	INT	The index for the array variable **LIFE( )** corresponding to the user-requested simulated failure distribution B-lives contained in variable **BLFPER( )**.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
CEE	RE	C in Equation 2-7, the Generalized Forman model parameter.
CI	RE	Initial crack size c_i (in.) for the elliptic surface flaw.
CO	RE	C_o in Equation 2-10, threshold stress intensity factor (SIF) ΔK_{TH} model parameter.
COEXP	RE	α (/ °R) in Equation 2-3, the COefficient of thermal EXPansion.
DADB(2)	RE	Block growth rate da/dB in the "a" and "c" directions.
DADN(MAXDIV, MAXDAT)	RE	2-D array containing the crack growth rate (in./cycle) in the da/dN vs. ΔK data.
DEE	RE	d in Equation 2-10, threshold SIF ΔK_{TH} model parameter.
DELK(MAXDIV, MAXDAT)	RE	2-D array containing the SIF range (ksi $\sqrt{\text{in.}}$) in the da/dN vs. ΔK data.
DESCRP	CH40	Description of the material.
DK	RE	SIF range ΔK(ksi $\sqrt{\text{in.}}$).
DKEFF	RE	Effective SIF range ΔK_{eff} after retardation given in Equation 2-16.
DKTH	RE	Threshold SIF range ΔK_{TH} (ksi $\sqrt{\text{in.}}$).
DKTHO	RE	Threshold SIF range (ksi $\sqrt{\text{in.}}$) at $R = 0$ used in Equation 2-10.
DLTAT	RE	DeLTA T. ΔT (°R) in Equation 2-2, the temperature difference across the wall of the duct.
DPCMU	RE	Value of (**PCMUB** – **PCMUA**).
DPCSIG	RE	Value of (**PCSIGB** – **PCSIGA**).
DSALT	RE	Bin stress interval for the stress level vs. number of cycles table from rainflow counting.
DSTR	RE	$\lambda_{DYN_{str}}$ in Equation 2-5, the randomly selected dynamic stress analysis accuracy factor.
DSTRA	RE	Uniform distribution lower bound of the dynamic stress analysis accuracy factor.
DSTRB	RE	Uniform distribution upper bound of the dynamic stress analysis accuracy factor.
DTIMU	RE	Value of (**TIMUB** – **TIMUA**).

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
DTISIG	RE	Value of (**TISIGB** – **TISIGA**).
DTOMU	RE	Value of (**TOMUB** – **TOMUA**).
DTOSIG	RE	Value of (**TOSIGB** – **TOSIGA**).
E(MAXSEG)	RE	1-D array containing the strain ε values for the stress/strain versus strain curve.
EM	RE	E (psi) in Equation 2-2, Young's modulus of elasticity for the material.
EMM	RE	m in Equation 2-7, the Generalized Forman model parameter.
ENN	RE	n in Equation 2-7, the Generalized Forman model parameter.
FAIL	LOG	Unstable crack growth indicator when $K > K_{cr}$.
FILNUM(MAXLD)	INT	1-D array containing the file unit numbers for the reference time history files.
FK(10)	RE	1-D array containing values of F_K, Equation 2-3, used to find stress concentration due to weld eccentricity, K_{OFF}.
FTEST	LOG	Used to test for existence of files.
FTY	RE	Material yield strength (psi).
INDIA	RE	D_i (in.), the randomly selected inner diameter.
INDIAA	RE	Lower bound of the Beta distribution on D_i.
INDIAB	RE	Upper bound of the Beta distribution on D_i.
INDIR	RE	Randomly selected Beta distribution location parameter ρ for D_i.
INDIR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution for D_i.
INDIR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution for D_i.
INDIT	RE	Randomly selected Beta distribution location parameter θ for D_i.
INDIT1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution for D_i.
INDIT2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution for D_i.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
INEUB	INT	Neuber's rule controller. **INEUB** = 0, no Neuber's equivalent mean stress calculation; **INEUB** = 1, include Neuber's equivalent mean stress calculation.
IOUT	INT	Output dump controller. **IOUT** = 0, no intermediate calculation output; **IOUT** = 15, driver sampling and driver transformation calculations; **IOUT** = 20, crack growth calculations; **IOUT** = 25, stress calculations; **IOUT** = 30, rainflow cycle counting.
IREGOP	INT	Regression options for Forman growth rate Equation 2-7. **IREGOP** = 0, fix p regress for C, n, m, q; **IREGOP** = 1, fix m, p regress for C, n, q; **IREGOP** = 2, fix q, p regress for C, n, m; **IREGOP** = 3, fix m, q, p regress for C, n; **IREGOP** = 4, regress for C, n, m, q, p.
IRET	INT	Willenborg's retardation model controller. **IRET** = 0, no growth retardation; **IRET** = 1, include growth retardation.
KC	RE	Critical stress intensity factor K_c (ksi).
KGROW	INT	Generalized Forman coefficient m controller. **KGROW** = 1, no m variation will be included; **KGROW** = 2, allows Uniform variation in m.
KLAM	RE	Randomly selected stress intensity factor calculation accuracy λ_{sif}.
KLAMA	RE	Uniform distribution lower bound of the stress intensity factor calculation accuracy.
KLAMB	RE	Uniform distribution upper bound of the stress intensity factor calculation accuracy.
KMAX(2)	RE	Maximum stress intensity factor K_{max} (ksi).
KMAXEF	RE	Effective maximum stress intensity factor $K_{max.eff}$ after retardation given in Equation 2-12.
KMIN(2)	RE	Minimum stress intensity factor K_{min} (ksi).
KMINEF	RE	Effective minimum stress intensity factor $K_{min.eff}$ after retardation given in Equation 2-12.
KOFF	RE	K_{OFF} in Equation 2-3, the stress concentration factor due to eccentricity of the weld.
KPROB	INT	Type of crack growth problem. **KPROB** = 1, analyze the HEX coil problem; **KPROB** = 2, analyze EXHEX problem.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LAMGR	RE	λ_{gro} in Equation 2-18, the randomly selected crack growth accuracy factor. See Section 2.2.4 for a discussion of crack growth calculations.
LAMGRA	RE	Uniform distribution lower bound of the crack growth accuracy factor.
LAMGRB	RE	Uniform distribution upper bound of the crack growth accuracy factor.
LAMKC	RE	λ_{K_c} in Equation 2-8, the randomly selected critical stress intensity factor uncertainty.
LAMKCA	RE	Uniform distribution lower bound of the critical stress intensity factor uncertainty.
LAMKCB	RE	Uniform distribution upper bound of the critical stress intensity factor uncertainty.
LAMKH	RE	$\lambda_{K_{TH}}$ in Equation 2-8, the randomly selected threshold stress intensity factor range uncertainty.
LAMKHA	RE	Uniform distribution lower bound of the threshold stress intensity factor range uncertainty.
LAMKHB	RE	Uniform distribution upper bound of the threshold stress intensity factor range uncertainty.
LAMN	RE	$\lambda_{D_{RANDOM}}$ in Equation 2-5, the randomly selected load scale factor for the narrow-band random loads. See Section 2.1.3.2 of [1] for a description of the parameters k, coefficient of variation C, and strain gage factor d.
LAMNA	RE	Lower bound of the Uniform distribution of k for the narrow-band random load scale factor.
LAMNB	RE	Upper bound of the Uniform distribution of k for the narrow-band random load scale factor.
LAMNC	RE	Coefficient of variation C for the narrow-band random load scale factor.
LAMND	RE	Strain gage correction factor d for the narrow-band random load scale factor.
LAMNK	RE	Randomly selected k for the narrow-band random load scale factor.
LAMNMU	RE	The resulting mean, μ, of the Normal distribution for the narrow-band random load scale factor, where $\mu = d/(1 + kC)$.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LAMNSG	RE	The resulting standard deviation, σ, of the Normal distribution for the narrow-band random load scale factor, where $\sigma = C/(1 + kC)$.
LAMS	RE	$\lambda_{DSINUSOIDAL}$ in Equation 2-5, the randomly selected load scale factor for the superimposed sinusoidal loads. See Section 2.1.3.2 of [1] for a description of the parameters k; coefficient of variation C; and strain gage factor d.
LAMSA	RE	Lower bound of the Uniform distribution of k for the superimposed sinusoidal load scale factor.
LAMSB	RE	Upper bound of the Uniform distribution of k for the superimposed sinusoidal load scale factor.
LAMSC	RE	Coefficient of variation C for the superimposed sinusoidal load scale factor.
LAMSD	RE	Strain gage correction factor d for the superimposed sinusoidal load scale factor.
LAMSK	RE	Randomly selected k for the superimposed sinusoidal load scale factor.
LAMSMU	RE	The resulting mean, μ, of the Normal distribution for the superimposed sinusoidal load scale factor, where $\mu = d/(1 + kC)$.
LAMSSG	RE	The resulting standard deviation, σ, of the Normal distribution for the superimposed sinusoidal load scale factor, where $\sigma = C/(1 + kC)$.
LAMW	RE	LAMbda Weld offset, the randomly selected λ_{OFF} in Equation 2-3, the accuracy factor for the weld offset eccentricity stress concentration factor, K_{OFF}.
LAMWA	RE	Uniform distribution lower bound of λ_{OFF}.
LAMWB	RE	Uniform distribution upper bound of λ_{OFF}.
LDNAME(MAXLD)	CH6	1-D array containing LoaD NAMEs for the dynamic or time-varying loads. These are the names of the reference time history files.
LIFE(MAXLIF)	RE	1-D array containing values of the lives generated by program PROCRK. The lives are sorted values for the left-hand tail simulated failure distribution.
LOCAT	INT	Critical location of interest on the HEX coil wall where 1 is the exterior surface of the duct, and 2 is the interior surface of the duct.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
M(2, MAXLD)	RE	2-D array containing the dynamic or time-varying moment load components. **M(1,*)** is M_y (in.-lbs) in Equation 2-1, the moment load components about the y-axis; and **M(2,*)** is M_z (in.-lbs) in Equation 2-1, the moment load components about the z-axis.
MAXBLF	INT	Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is 10.
MAXDAT	INT	Maximum number of points per data division allowed for da/dN vs. ΔK curve. The maximum number of data points per division allowed is 200.
MAXDIV	INT	Maximum number of data divisions allowed for da/dN vs. ΔK curve. The maximum number of data divisions allowed is 10.
MAXLD	INT	Maximum number of dynamic or time-varying loads allowed. The maximum number of loads is 16.
MAXLIF	INT	Maximum number of crack growth lives allowed for the simulated failure distribution. The maximum number of crack growth lives to be saved is 1000.
MAXM	INT	Maximum number of points allowed in the time history arrays. The maximum number of points is 20,000.
MAXSEG	INT	Maximum number of segments allowed in the stress-strain versus strain curve. The maximum number of segments is 10.
MI	RE	I (in.^{4}) in Equation 2-1, the cross-sectional Moment of Inertia.
MLAM(2, MAXLD)	RE	2-D array containing the dynamic or time-varying moment load components scaled by **DSTR** or **ASTR** and **LAMS**, **LAMN**, or **AERD**, as appropriate, according to variable **TYPE( )**. **MLAM(1,*)** is M_y (in.-lbs) in Equation 2-1, the moment load components about the y-axis; and **MLAM(2,*)** is M_z (in.-lbs) in Equation 2-1, the moment load components about the z-axis.
MSLAM(2)	RE	1-D array containing the static moment load components scaled by **ASTR**, and **AERS**, or **SSTR** as appropriate. **MSLAM(1)** is M_y (in.-lbs) in Equation 2-1, the moment load component about the y-axis;

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
		and **MSLAM(2)** is M_z (in.-lbs) in Equation 2-1, the moment load component about the z-axis.
MSTAT(2)	RE	1-D array containing the static moment load components. **MSTAT(1)** is M_y (in.-lbs) in Equation 2-1, the moment load component about the y-axis; and **MSTAT(2)** is M_z (in.-lbs) in Equation 2-1, the moment load component about the z-axis.
MVAR	RE	Randomly selected Forman coefficient *m*.
MVARA	RE	Uniform distribution lower bound of the Forman coefficient *m*.
MVARB	RE	Uniform distribution upper bound of the Forman coefficient *m*.
NBIN(100)	INT	1-D array containing the number of cycles for the stress level vs. number of cycles table from rainflow counting.
NBLIFE	INT	Number of B-lives to be obtained from the simulated failure distribution.
NCRL	INT	Number of crack lengths for life calculations.
NDIR	INT	Number of directions to grow the crack in.
NDIV	INT	Number of crack growth data divisions.
NEUB	RE	Randomly selected Neuber's rule model accuracy factor λ_{neu}.
NEUBA	RE	Uniform distribution lower bound of the Neuber's rule model accuracy factor.
NEUBB	RE	Uniform distribution upper bound of the Neuber's rule model accuracy factor.
NEWLIF	INT	Crack growth life value returned from call to LIFCAL.
NHYPER	INT	The outer loop size.
NLIFE	INT	The inner loop size.
NLIFET	INT	Total number of lives calculated by program PROCRK. Value of **NHYPER * NLIFE**.
NLOAD	INT	NLOAD in Equation 2-5, the number of dynamic or time-varying loads.
NP(MAXDIV)	INT	1-D array containing the number of points per data division for the material *da/dN* data set.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
NRAN	INT	Number of RANdom points. Number of points in the reference time history.
NU	RE	ν in Equation 2-2, the materials Poisson's ratio.
NUMSEG	INT	Number of segments of interest in stress-strain versus strain curve.
P(MAXLD)	RE	1-D array containing P (lbs) in Equation 2-1, the dynamic or time-varying axial load components.
PC	RE	p_i (psi) in Equation 2-1, the randomly selected internal pressure.
PCMU	RE	Randomly selected Normal distribution parameter μ for the internal pressure p_i.
PCMUA	RE	Uniform distribution lower bound of parameter μ in the Normal distribution of the internal pressure p_i.
PCMUB	RE	Uniform distribution upper bound of parameter μ in the Normal distribution of the internal pressure p_i.
PCO	RE	p_o (psi) in Equation 2-1, the external pressure.
PCSIG	RE	Randomly selected Normal distribution parameter σ for the internal pressure p_i.
PCSIGA	RE	Uniform distribution lower bound of parameter σ in the Normal distribution of the internal pressure p_i.
PCSIGB	RE	Uniform distribution upper bound of parameter σ in the Normal distribution of the internal pressure p_i.
PEE	RE	p in Equation 2-7, the Generalized Forman model parameter.
PERIOD	RE	T (sec) in Equation 2-18, the length of time in seconds of the reference time history.
PI	RE	π, constant equal to 3.1415926536...
PLAM(MAXLD)	RE	1-D array containing P (lbs) in Equation 2-1, the dynamic or time-varying axial load components scaled by **DSTR** or **ASTR** and **LAMN**, **LAMS**, or **AERD**, as appropriate, according to variable **TYPE( )**.
PSLAM	RE	P (lbs) in Equation 2-1, the static axial load component scaled by **ASTR**, and **AERS**, or **SSTR** as appropriate.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
PSTAT	RE	P (lbs) in Equation 2-1, the static axial load component.
QUE	RE	q in Equation 2-7, the Generalized Forman model parameter.
RAND	DRE	Random number seed.
RDATA(MAXDIV)	RE	1-D array containing the stress ratio for growth rate data for each data division.
REFF	RE	Effective stress ratio $K_{min.eff}/K_{max.eff}$ after retardation given by Equation 2-16.
RI	RE	R_i (in.) in Equation 2-1, the duct inner radius.
RO	RE	R_o (in.) in Equation 2-1, the duct outer radius.
ROT	RE	R Over T, the value of the ratio R/t.
RSO	RE	Willenborg retardation model parameter as given in Equation 2-13.
RT(10)	RE	1-D array containing values of R/t used in conjunction with F_k, Equation 2-3, to find stress concentration due to weld eccentricity, K_{OFF}.
SE(MAXSEG)	RE	1-D array containing values of the product of stress and strain $\sigma\varepsilon$ for each segment of the stress-strain versus strain curve.
SPR(MAXM)	RE	1-D array containing the principal stress-time history $\sigma(t)$ (psi), Equation 2-5, resulting from the combination of stresses from static, narrow-band random, superimposed sinusoidal, and aerodynamic load sources.
SSTR	RE	λST_{str} in Equation 2-5, the randomly selected static stress analysis accuracy factor.
SSTRA	RE	Uniform distribution lower bound of the static stress analysis accuracy factor.
SSTRB	RE	Uniform distribution upper bound of the static stress analysis accuracy factor.
STATIC(4)	RE	1-D array containing values of the static stresses σ_{ST} (psi), Equation 2-5. **STATIC(1)** is the axial stress σ_{ST}. **STATIC(2)**, **STATIC(3)**, and **STATIC(4)** are not used in the HEX coil or EXHEX analyses.
STRAMP(4, MAXLD)	RE	2-D array containing values of the amplitudes of the dynamic or time-varying stresses $\overline{\sigma_{Di}}$ (psi),

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
		Equation 2-5. **STRAMP(1,I)** is $\overline{\sigma_{Di}}$, the amplitude of the *i*th axial stress. **STRAMP(2,I)**, **STRAMP(3,I)**, and **STRAMP(4,I)** are not used in the HEX coil or EXHEX analyses.
STRHIS(MAXLD, MAXM)	RE	2-D array containing $\sigma_i(t)$, Equation 2-5, the reference time histories for the dynamic or time-varying load components.
SX(MAXLD)	RE	1-D array containing the time-varying magnitude of σ_x (psi) stress component.
SXY(MAXLD)	RE	1-D array containing the time-varying magnitude of σ_{xy} (psi) stress component.
SXZ(MAXLD)	RE	1-D array containing the time-varying magnitude of σ_{xz} (psi) stress component.
SXST	RE	Static σ_x (psi) stress component.
SXYST	RE	Static σ_{xy} (psi) stress component.
SXZST	RE	Static σ_{xz} (psi) stress component.
SY(MAXLD)	RE	1-D array containing the time-varying magnitude of σ_y (psi) stress component.
SYZ(MAXLD)	RE	1-D array containing the time-varying magnitude of σ_{yz} (psi) stress component.
SZ(MAXLD)	RE	1-D array containing the time-varying magnitude of σ_z (psi) stress component.
SYST	RE	Static σ_y (psi) stress component.
SYZST	RE	Static σ_{yz} (psi) stress component.
SZST	RE	Static σ_z (psi) stress component.
T(MAXLD)	RE	1-D array containing M_x (in.-lbs) the dynamic or time-varying torsional load components. Not used in the HEX coil or EXHEX analysis.
THIC	RE	t (in.), the randomly selected wall thickness at the weld used to calculate the area A and outer radius R_o in Equation 2-1.
THICA	RE	Lower bound of the Beta distribution on t.
THICB	RE	Upper bound of the Beta distribution on t.
THICR	RE	Randomly selected Beta distribution location parameter ρ for the wall thickness t.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
THICR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution of t.
THICR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution of t.
THICT	RE	Randomly selected Beta distribution location parameter θ for the wall thickness t.
THICT1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution of t.
THICT2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution of t.
TIN	RE	T_i (°R), the randomly selected inner wall surface temperature, used to calculate ΔT (°R), the temperature difference across the wall of the duct, given in Equation 2-2.
TIMU	RE	Randomly selected Normal distribution parameter μ for the inner wall surface temperature T_i.
TIMUA	RE	Uniform distribution lower bound of parameter μ in the Normal distribution of the inner wall surface temperature T_i.
TIMUB	RE	Uniform distribution upper bound of parameter μ in the Normal distribution of the inner wall surface temperature T_i.
TISIG	RE	Randomly selected Normal distribution parameter σ for the inner wall surface temperature T_i.
TISIGA	RE	Uniform distribution lower bound of parameter σ in the Normal distribution of the inner wall surface temperature T_i.
TISIGB	RE	Uniform distribution upper bound of parameter σ in the Normal distribution of the inner wall surface temperature T_i.
TOUT	RE	T_o (°R), the randomly selected outer wall surface temperature, used to calculate ΔT (°R), the temperature difference across the wall of the duct, given in Equation 2-2.
TOMU	RE	Randomly selected Normal distribution parameter μ for the outer wall surface temperature T_o.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
TOMUA	RE	Uniform distribution lower bound of parameter μ in the Normal distribution of the outer wall surface temperature T_o.
TOMUB	RE	Uniform distribution upper bound of parameter μ in the Normal distribution of the outer wall surface temperature T_o.
TOSIG	RE	Randomly selected Normal distribution parameter σ for the outer wall surface temperature T_o.
TOSIGA	RE	Uniform distribution lower bound of parameter σ in the Normal distribution of the outer wall surface temperature T_o.
TOSIGB	RE	Uniform distribution upper bound of parameter σ in the Normal distribution of the outer wall surface temperature T_o.
TRUNC	RE	Value used to filter out noise in the principal stress-time history during rainflow cycle counting. See Section 2.2.1.4 of [1] for a discussion of rainflow cycle counting.
TSTAT	RE	M_x (in.-lbs), the static torsional load component. Not used in the HEX coil or EXHEX analysis.
TYPE(MAXLD)	INT	1-D array containing the type of dynamic or time-varying load, used to assign the appropriate load scale factors. **TYPE(*) = 1**, use the narrow-band random load scale factor; **TYPE(*)** = 2, use the superimposed sinusoidal load scale factor; and **TYPE(*)** = 3, use the aerodynamic load factor.
V(2, MAXLD)	RE	2-D array containing the time-varying shear load components V_y and V_z (lbs). Not used in the HEX coil or EXHEX analysis.
VSTAT(2)	RE	1-D array containing the static shear load components V_y and V_z (lbs). Not used in the HEX coil or EXHEX analysis.
WIDTH	RE	W (in.), the randomly selected plate width used to calculate the SIF for the EXHEX crack configuration.
WITHA	RE	Lower bound of Beta distribution for W.
WITHB	RE	Upper bound of Beta distribution for W.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
WITHR	RE	Randomly selected Beta distribution location parameter ρ for the width W.
WITHR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution of W.
WITHR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution of W.
WITHT	RE	Randomly selected Beta distribution location parameter θ for the width W.
WITHT1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution of W.
WITHT2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution of W.
WOFF	RE	W_{OFF} in Equation 2-3, the randomly selected Weld OFFset (%).
WOFFA	RE	Lower bound of the first Beta distribution on W_{OFF}.
WOFFB	RE	Upper bound of the first Beta distribution on W_{OFF}.
WOFFC	RE	Lower bound of the second Beta distribution on W_{OFF}.
WOFFD	RE	Upper bound of the second Beta distribution on W_{OFF}.
WOFFE	RE	Decimal equivalent percentage weight occurring in the first Beta distribution of the weld offset W_{OFF}.
WOFFHI	RE	Upper bound of the randomly selected Beta distribution for the weld offset W_{OFF}.
WOFFLO	RE	Lower bound of the randomly selected Beta distribution for the weld offset W_{OFF}.
WOFFR	RE	Randomly selected Beta distribution location parameter ρ for the weld offset W_{OFF}.
WOFFR1	RE	Uniform distribution lower bound of parameter ρ in the first Beta distribution of W_{OFF}.
WOFFR2	RE	Uniform distribution upper bound of parameter ρ in the first Beta distribution of W_{OFF}.
WOFFR3	RE	Uniform distribution lower bound of parameter ρ in the second Beta distribution of W_{OFF}.
WOFFR4	RE	Uniform distribution upper bound of parameter ρ in the second Beta distribution of W_{OFF}.

Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
WOFFT	RE	Randomly selected Beta distribution shape parameter θ for the weld offset W_{OFF}.
WOFFT1	RE	Uniform distribution lower bound of parameter θ in the first Beta distribution of W_{OFF}.
WOFFT2	RE	Uniform distribution upper bound of parameter θ in the first Beta distribution of W_{OFF}.
WOFFT3	RE	Uniform distribution lower bound of parameter θ in the second Beta distribution of W_{OFF}.
WOFFT4	RE	Uniform distribution upper bound of parameter θ in the second Beta distribution of W_{OFF}.

7.1.4 Program PROCRK Listing

PROCRK Version 92.5

Program PROCRK Listing Temporal Order

```
C***********************************************************************
C  PROCRK IS THE MAIN MODULE OF THE PROBABILISTIC CRACK GROWTH PROGRAM
C  PROGRAMMER:  S. SUTHARSHANA
C
C  THIS PROGRAM DRAWS MANY ROUTINES FROM PROGRAM HEXHCF (JPL PUB 92-15)
C
C  DATE:  DECEMBER 1992
C  VERSION: 92.5
C
C  Copyright (C) 1991, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************
      PROGRAM PROCRK
C=======================================================================
C  SUBPROGRAMS:  SETDEF, INPUT, GRODAT, HYPDRW, PARDRW, LIFCAL, INSORT
C
C        FILES:  1:CRKDAT-OLD; 3:CRKRES-NEW; 8:IOUTPR-NEW;
C                9:LOWLIF-NEW; 11-26:user named-OLD
C
C=======================================================================
C       IMPLICIT NONE

      INTEGER  MAXBLF, MAXLD, MAXLIF, MAXM, MAXSEG

      PARAMETER (MAXBLF = 10, MAXLD = 16, MAXLIF = 1000,
     &           MAXM = 20000, MAXSEG = 10)

      INTEGER  BLFPOS, I, IOUT, J, K, LOCAT, NBLIFE,  NDIR,
     &         NCRL, NHYPER, NLIFE, NLIFET, NLOAD, NRAN,
     &         NUM, NUMSEG, TYPE(MAXLD)

      INTEGER INEUB, IRET, KGROW, KPROB

      DOUBLE PRECISION  RAND

      REAL     AERD, AERDA, AERDB, AERS, AERSA, AERSB, AI, AIA,
     &         AIB, AIR, AIR1, AIR2, AIT,
     &         AIT1, AIT2, ANGLE, AOCA, AOCB, AOCR, AOCR1,
     &         AOCR2, AOCT, AOCT1, AOCT2, ASTR, ASTRA, ASTRB,
     &         BLFPER(MAXBLF), CEE, CI, CO,
     &         COEXP, DEE,
     &         DKTHO, DLTAT, DPCMU, DPCSIG, DSTR, DSTRA, DSTRB,
     &         DTIMU, DTISIG, DTOMU, DTOSIG, E(MAXSEG), EM,
     &         EMM, ENN, FTY,
     &         INDIA, INDIAA, INDIAB, INDIR, INDIR1, INDIR2,
     &         INDIT, INDIT1, INDIT2, KC,
     &         KLAM, KLAMA, KLAMB, LAMKH, LAMKHA, LAMKHB,
     &         LAMKC, LAMKCA, LAMKCB, LAMGR, LAMGRA, LAMGRB,
     &         LAMN, LAMNA, LAMNB, LAMNC, LAMND, LAMNMU, LAMNSG,
     &         LAMS, LAMSA, LAMSB, LAMSC, LAMSD, LAMSMU, LAMSSG,
     &         LAMW, LAMWA, LAMWB, LIFE(MAXLIF)

      REAL     M(2, MAXLD), MSTAT(2),
     &         MVAR, MVARA, MVARB, NEUB, NEUBA, NEUBB, NEWLIF, NU,
     &         P(MAXLD), PC, PCMU, PCMUA, PCMUB, PCO, PCSIG, PCSIGA,
     &         PCSIGB, PEE, PERIOD, PSTAT, QUE, RSO,
     &         SE(MAXSEG), SSTR, SSTRA,
     &         SSTRB, STRHIS(MAXLD, MAXM), SX(MAXLD), SXST,
     &         SXY(MAXLD), SXYST, SXZ(MAXLD), SXZST, SY(MAXLD),
     &         SYST, SYZ(MAXLD), SYZST, SZ(MAXLD), SZST,
     &         T(MAXLD), THIC, THICA, THICB, THICR, THICR1,
     &         THICR2, THICT, THICT1, THICT2, TIMU, TIMUA,
     &         TIMUB, TISIG, TISIGA, TISIGB, TOMU,
     &         TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
     &         TRUNC, TSTAT, V(2, MAXLD),
     &         VSTAT(2), WIDTH, WITHA,
     &         WITHB, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
     &         WOFF, WOFFA, WOFFB, WOFFC, WOFFD,
     &         WOFFE, WOFFHI, WOFFLO, WOFFR, WOFFR1, WOFFR2, WOFFR3,
     &         WOFFR4, WOFFT, WOFFT1, WOFFT2, WOFFT3, WOFFT4

      CHARACTER*6 LDNAME(MAXLD)
```

```
      COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
     &              P, T, M, V, PCO, SXST, SYST, SZST, SXYST,
     &              SXZST, SYZST, SX, SY, SZ, SXY, SXZ, SYZ

      COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AIA,
     &    AIB, AIR1, AIR2, AIT1, AIT2, AIR, AIT,
     &    AOCA, AOCB, AOCR1, AOCR2, AOCT1, AOCT2, AOCR, AOCT,
     &    ASTRA, ASTRB,
     &    DPCMU, DPCSIG, DSTRA, DSTRB, DTIMU, DTISIG, DTOMU, DTOSIG,
     &    INDIAA,INDIAB,INDIR1, INDIR2, INDIR, INDIT1, INDIT2, INDIT,
     &    KLAMA, KLAMB, LAMGRA, LAMGRB, LAMKHA, LAMKHB, LAMKCA, LAMKCB,
     &    LAMNA, LAMNB, LAMNC,  LAMND, LAMNMU, LAMNSG,
     &    LAMSA, LAMSB, LAMSC,  LAMSD, LAMSMU, LAMSSG,
     &    LAMWA, LAMWB, MVARA, MVARB, NEUBA,  NEUBB,
     &    PCMU, PCMUA, PCMUB, PCSIG, PCSIGA, PCSIGB,
     &    RAND,
     &    SSTRA, SSTRB,
     &    THICA, THICB, THICR1, THICR2, THICR, THICT1, THICT2, THICT,
     &    TIMU, TIMUA, TIMUB, TISIG, TISIGA, TISIGB,
     &    TOMU, TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
     &    WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
     &    WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
     &    WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFR, WOFFT1, WOFFT2,
     &    WOFFT3, WOFFT4, WOFFT

      COMMON/CNTRL/INEUB, IRET, KGROW, KPROB

      COMMON/NAMES/LDNAME

      COMMON IOUT

C***********************************************************************
C                       IMPORTANT VARIABLES IN PROCRK
C***********************************************************************
C
C AERD          ACCURACY FACTOR FOR AERO DYNAMIC LOADS
C AERDA         AERO DYNAMIC LOADS ACCURACY UNIFORM DISTRIBUTION LOWER BOUND
C AERDB         AERO DYNAMIC LOADS ACCURACY UNIFORM DISTRIBUTION UPPER BOUND
C AERS          ACCURACY FACTOR FOR AERO STATIC LOADS
C AERSA         AERO STATIC LOADS ACCURACY UNIFORM DISTRIBUTION LOWER BOUND
C AERSB         AERO STATIC LOADS ACCURACY UNIFORM DISTRIBUTION UPPER BOUND
C AI            INITIAL CRACK DIMENSION "a"
C AIA           INITIAL CRACK LOWER BOUND
C AIB           INITIAL CRACK UPPER BOUND
C AIR           INITIAL CRACK CHOSEN RHO                 BETA HYPER-
C AIR1          INITIAL CRACK RHO LOWER BOUND            DISTRIBUTION
C AIR2          INITIAL CRACK RHO UPPER BOUND            PARAMETERS
C AIT           INITIAL CRACK CHOSEN THETA
C AIT1          INITIAL CRACK THETA LOWER BOUND
C AIT2          INITIAL CRACK THETA UPPER BOUND
C ANGLE         ANGLE THETA MEASURED COUNTER-CLOCKWISE FROM THE Z-DIRECTION
C                 GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
C AOC           INITIAL SHAPE "a/c"
C AOCA          INITIAL SHAPE LOWER BOUND
C AOCB          INITIAL SHAPE UPPER BOUND
C AOCR          INITIAL SHAPE CHOSEN RHO                 BETA HYPER-
C AOCR1         INITIAL SHAPE RHO LOWER BOUND            DISTRIBUTION
C AOCR2         INITIAL SHAPE RHO UPPER BOUND            PARAMETERS
C AOCT          INITIAL SHAPE CHOSEN THETA
C AOCT1         INITIAL SHAPE THETA LOWER BOUND
C AOCT2         INITIAL SHAPE THETA UPPER BOUND
C ASTR          AERODYNAMIC STRESS ANALYSIS ACCURACY
C ASTRA         AERODYNAMIC STRESS ANALYSIS ACCURACY UNIFORM DISTRIBUTION LOWER
C                 BOUND
C ASTRB         AERODYNAMIC STRESS ANALYSIS ACCURACY UNIFORM DISTRIBUTION UPPER
C                 BOUND
C BLFPER()      1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE CALCULATED
C BLFPOS        POSITION IN LIFE() OF EMPIRICAL BLIVES
C CEE           COEFFICIENT "C" IN THE GENERALIZED FORMAN MODEL
C CI            INITIAL CRACK SIZE "c"
C CO            THRESHOLD MODEL COEFFICIENT "Co"
C COEXP         COEFFICIENT OF THERMAL EXPANSION
C DADB(2)       BLOCK GROWTH RATE da/dB (1=a DIRECTION, 2=c DIRECTION)
C DADN()        GROWTH RATE DATA ARRAY da/dN
```

```
C DEE           THRESHOLD MODEL COEFFICIENT "d"
C DELK()        SIF RANGE FOR GROWTH RATE DATA
C DESCRP        DESCRIPTION OF THE MATERIAL DATA
C DK            SIF RANGE
C DKEFF         EFFECTIVE SIF RANGE AFTER RETARDATION
C DKTH          THRESHOLD SIF RANGE
C DKTHO         THRESHOLD MODEL COEFFICIENT "DKtho"
C DLTAT         SELECTED TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER
C                 SURFACES -- DeLTA (Ti - To)
C DPCMU         EQUAL TO PCMUB - PCMUA
C DPCSIG        EQUAL TO PCSIGB - PCSIGA
C DSALT         BIN STRESS INTERVAL FOR STRESS LEVEL VS. NUMBER OF CYCLES TABLE
C DSTR          SELECTED DYNAMIC STRESS ANALYSIS ACCURACY FACTOR
C DSTRA         DYNAMIC STRESS ANALYSIS ACCURACY FACTOR LOWER BOUND
C DSTRB         DYNAMIC STRESS ANALYSIS ACCURACY FACTOR UPPER BOUND
C DTIMU         EQUAL TO TIMUB - TIMUA
C DTISIG        EQUAL TO TISIGB - TISIGA
C DTOMU         EQUAL TO TOMUB - TOMUA
C DTOSIG        EQUAL TO TOSIGB - TOSIGA
C E()           1-D ARRAY WHICH CONTAINS THE STRAIN VALUES
C EM            ELASTIC MODULUS
C EMM           COEFFICIENT "m" IN THE GENERALIZED FORMAN MODEL
C ENN           COEFFICIENT "n" IN THE GENERALIZED FORMAN MODEL
C FAIL          LOGICAL VARIABLE TO INDICATE UNSTABLE CRACK,
C                 K GREATER THAN Kcr
C FILNUM()      1-D ARRAY CONTAINING UNIT NUMBER FOR STRESS-TIME HISTORIES
C                 FILES
C FK(10)        1-D ARRAY WITH Fk VALUES OF THE Fk VS. Rt CURVE
C FTEST         File TEST -- USED TO TEST EXISTENCE OF FILE
C FTY           MATERIAL YIELD STRENGTH
C INDIA         SELECTED INTERIOR DIAMETER                  \
C INDIAA        INTERIOR DIAMETER LOWER BOUND                |
C INDIAB        INTERIOR DIAMETER UPPER BOUND                |
C INDIR         SELECTED RHO FOR INTERIOR DIAMETER           |  BETA HYPER-
C INDIR1        INTERIOR DIAMETER - RHO LOWER BOUND           \ DISTRIBUTION
C INDIR2        INTERIOR DIAMETER - RHO UPPER BOUND           / PARAMETERS
C INDIT         SELECTED THETA FOR INTERIOR DIAMETER         |
C INDIT1        INTERIOR DIAMETER - THETA LOWER BOUND        |
C INDIT2        INTERIOR DIAMETER - THETA UPPER BOUND       /
C INEUB         NEUBER'S RULE CONTROL (1=INCLUDE, 0=EXCLUDE)
C IOUT          CONTROLS DUMP TO SCREEN/PRINTER
C IRET          WILLENBORG'S RETARDATION MODEL CONTROLLER (1=INCLUDE,
C                 0=EXCLUDE)
C IREGOP        FORMAN EQUATION REGRESSION OPTION
C KC            CRITICAL STRESS INTENSITY FACTOR Kc
C KGROW         GROWTH MODEL, GENERALIZED FORMAN COEFFICINET m (CONST=1,
C                 VARY=2)
C KLAM          STRESS INTENSITY FACTOR CALCULATION ACCURACY
C KLAMA         SIF CALCULATION ACCURACY UNIFORM DISTRIBUTION LOWER BOUND
C KLAMB         SIF CALCULATION ACCURACY UNIFORM DISTRIBUTION UPPER BOUND
C KMAX()        MAXIMUM SIF
C KMAXEF        EFFECTIVE MAXIMUM SIF AFTER RETARDATION
C KMIN()        MINIMUM SIF
C KMINEF        EFFECTIVE MINIMUM SIF AFTER RETARDATION
C KOFF          STRESS CONCENTRATION DUE TO WELD OFFSET
C KPROB         TYPE OF PROBLEM (HEX COIL = 1, EXHEX = 2)
C LAMGR         GROWTH CALCULATION ACCURACY FACTOR
C LAMGRA        GROWTH CALCULATION ACCURACY UNIFORM DISTRIBUTION LOWER BOUND
C LAMGRB        GROWTH CALCULATION ACCURACY UNIFORM DISTRIBUTION UPPER BOUND
C LAMKC         CRITICAL SIF Kc UNCERTAINTY
C LAMKCA        CRITICAL SIF Kc UNCERTAINTY UNIFORM DISTRIBUTION LOWER BOUND
C LAMKCB        CRITICAL SIF Kc UNCERTAINTY UNIFORM DISTRIBUTION UPPER BOUND
C LAMKH         THRESHOLD SIF Kth UNCERTAINTY
C LAMKHA        THRESHOLD SIF Kth UNCERTAINTY UNIFORM DISTRIBUTION LOWER BOUND
C LAMKHB        THRESHOLD SIF Kth UNCERTAINTY UNIFORM DISTRIBUTION UPPER BOUND
C LAMN          SELECTED LAMBDA FOR ONE SIGMA NARROW-BAND RANDOM LOADS
C LAMNA         LAMBDA FOR NARROW-BAND RANDOM LOADS -- LOWER BOUND OF k
C LAMNB         LAMBDA FOR NARROW-BAND RANDOM LOADS -- UPPER BOUND OF k
C LAMNC         LAMBDA FOR NARROW-BAND RANDOM LOADS COEFFICIENT OF VARIATION
C LAMND         NARROW-BAND RANDOM LOADS STRAIN GAGE ACCURACY FACTOR
C LAMNK         LAMBDA FOR NARROW-BAND RANDOM LOADS k -- INDICATES VARIATION
C                 DUE TO SAMPLE SIZE
C LAMNMU        MEAN OF LAMBDA FOR NARROW-BAND RANDOM LOADS (MU, NORMAL
C                 DISTRIBUTION)
C LAMNSG        STANDARD DEVIATION OF LAMBDA FOR NARROW-BAND RANDOM LOADS
```

```
C                 (SIGMA, NORMAL DISTRIBUTION)
C LAMS           SELECTED LAMBDA FOR SUPERIMPOSED SINE LOADS
C LAMSA          LAMBDA FOR SUPERIMPOSED SINE LOADS -- LOWER BOUND OF k
C LAMSB          LAMBDA FOR SUPERIMPOSED SINE LOADS -- UPPER BOUND OF k
C LAMSC          LAMBDA FOR SUPERIMPOSED SINE LOADS COEFFICIENT OF VARIATION
C LAMND          SUPERIMPOSED SINE LOADS STRAIN GAGE ACCURACY FACTOR
C LAMSK          LAMBDA FOR SUPERIMPOSED SINE LOADS k -- INDICATES VARIATION
C                  DUE TO SAMPLE SIZE
C LAMSMU         MEAN OF LAMBDA FOR SUPERIMPOSED SINE LOADS (MU, NORMAL
C                  DISTRIBUTION)
C LAMSSG         STANDARD DEVIATION OF LAMBDA FOR SUPERIMPOSED SINE LOADS
C                  (SIGMA, NORMAL DISTRIBUTION)
C LAMW           SELECTED ACCURACY FACTOR FOR WELD ECCENTRICITY STRESS
C                  CONCENTRATION FACTOR, Koff
C LAMWA          LAMW LOWER BOUND
C LAMWB          LAMW UPPER BOUND
C LDNAME()       1-D ARRAY CONTAINING LoaD NAMEs FOR THE TIME-VARYING LOADS
C LIFE()         1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED
C                  -- SORTED VALUES OF THE LEFT-HAND TAIL
C LOCAT          LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE OF THE
C                  DUCT, AND 2 IS THE INTERIOR SURFACE OF THE DUCT
C M()            2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                  ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C MAXBLF         MAXIMUM NUMBER OF PERCENTAGE PROBABILITY LEVELS
C MAXDAT         MAXIMUM NUMBER OF POINTS PER DATA DIVISION ALLOWED
C MAXDIV         MAXIMUM NUMBER OF DATA DIVISIONS ALLOWED
C MAXLD          MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C MAXLIF         MAXIMUM NUMBER OF CRACK GROWTH LIVES ALLOWED
C MAXM           MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
C MAXSEG         MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN CURVE)
C MI             MOMENT OF INERTIA FOR DUCT
C MLAM()         2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS SCALED
C                  BY DSTR OR ASTR AND LAMS, LAMN, OR AERD AS APPROPRIATE
C                  (INDICATED BY TYPE()) -- MLAM(1,*) ARE THE M2 LOADS;
C                  MLAM(2,*) ARE THE M3 LOADS
C MSLAM()        1-D ARRAY CONTAINING THE STATIC LOADS SCALED BY ASTR
C                  AND AERS OR SSTR AS APPROPRIATE -- MSLAM(1) IS THE M2 LOAD;
C                  MSLAM(2) IS THE M3 LOAD
C MSTAT()        1-D ARRAY CONTAINING THE STATIC LOADS -- MSTAT(1) IS THE M2
C                  LOAD; MSTAT(2) IS THE M3 LOAD
C MVAR           SELECTED FORMAN COEFFICIENT m
C MVARA          FORMAN COEFFICIENT m UNIFORM DISTRIBUTION LOWER BOUND
C MVARB          FORMAN COEFFICIENT m UNIFORM DISTRIBUTION UPPER BOUND
C NBIN(100)      1-D ARRAY CONTAINING THE NUMBER OF CYCLES AFTER RF COUNTING
C NBLIFE         NUMBER OF BLIVES TO BE CALCULATED
C NCRL           NUMBER OF CRACK LENGTHS FROM Ai TO Af TO DO GROWTH INTEGRATION
C NDIR           NUMBER OF DEGRESS OF FREEDOM FOR CRACK GROWTH (1 OR 2)
C NDIV           NUMBER OF DIVISIONS OF GROWTH RATE DATA
C NEUB           SELECTED NEUBER'S RULE MODEL ACCURACY FACTOR
C NEUBA          NEUB UNIFORM DISTRIBUTION LOWER BOUND
C NEUBB          NEUB UNIFORM DISTRIBUTION UPPER BOUND
C NEWLIF         LIFE VALUE RETURNED FROM CALL TO LIFCAL
C NHYPER         NUMBER OF SETS OF HYPERPARAMETER DISTRIBUTIONS TO BE
C                  SAMPLED FROM
C NLIFE          NUMBER OF DUCT FAILURE LIVES TO BE CALCULATED
C NLIFET         TOTAL NUMBER OF LIVES CALCULATED
C NLOAD          NUMBER OF TIME-VARYING LOADS
C NORM           RANDOM VARIABLE (SOMETIMES UNIFORM, SOMETIMES NORMAL) USED
C                  TO OBTAIN SELECTED TEMPERATURES AND PRESSURE
C NP()           1-D ARRAY CONTAINING NUMBER OF POINTS PER DATA DIVISION
C                  FOR CRACK GROWTH RATE DATA
C NRAN           NUMBER OF POINTS IN STRESS-TIME HISTORY
C NU             POISSON'S RATIO
C NUMSEG         NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
C P()            1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C PC             SELECTED INTERNAL PRESSURE, PSI
C PCMU           SELECTED MEAN OF NORMALLY DISTRIBUTED INTERNAL PRESSURE
C PCMUA          MEAN OF INTERNAL PRESSURE LOWER BOUND
C PCMUB          MEAN OF INTERNAL PRESSURE UPPER BOUND
C PCO            EXTERNAL PRESSURE, PSI
C PCSIG          SELECTED STANDARD DEVIATION OF NORMALLY DISTRIBUTED
C                  INTERNAL PRESSURE
C PCSIGA         STANDARD DEVIATION OF INTERNAL PRESSURE LOWER BOUND
C PCSIGB         STANDARD DEVIATION OF INTERNAL PRESSURE UPPER BOUND
C PEE            COEFFICIENT "p" IN THE GENERALIZED FORMAN MODEL
```

```
C PERIOD        LENGTH OF TIME IN SECONDS OF STRESS-TIME HISTORY
C PI            CONSTANT FOR THE VALUE 3.14..
C PLAM()        1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS SCALED
C                 BY DSTR OR ASTR AND LAMS, LAMN, OR AERD,
C                 AS APPROPRIATE (INDICATED BY TYPE())
C PSLAM         STATIC AXIAL LOAD SCALED BY ASTR AND AERS OR SSTR AS
C                 APPROPRIATE
C PSTAT         STATIC AXIAL LOAD
C QUE           COEFFICIENT "q" IN THE GENERALIZED FORMAN MODEL
C RAND          RANDOM NUMBER SEED
C RDATA()       STRESS RATIO R FOR GROWTH RATE DATA
C REFF          EFFECTIVE STRESS RATIO AFTER RETARDATION
C RI            INNER RADIUS FOR DUCT
C RO            OUTSIDE RADIUS FOR DUCT
C ROT           RATIO R/t
C RSO           WILLENBORG RETARDATION MODEL CONSTANT
C RT(10)        1-D ARRAY CONTAINING THE R/t VALUES OF THE Fk vs. Rt CURVE
C SE()          1-D ARRAY OF PRODUCT OF STRESS AND STRAIN FOR EACH SEGMENT OF
C                 THE STRESS-STRAIN VS STRAIN CURVE
C SPR()         PRINCIPAL STRESS HISTORY (MAXM)
C SSTR          SELECTED STATIC STRESS ANALYSIS ACCURACY
C SSTRA         SSTR UNIFORM DISTRIBUTION LOWER BOUND
C SSTRB         SSTR UNIFORM DISTRIBUTION UPPER BOUND
C STATIC()      1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES
C STRAMP()      2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C STRHIS()      2-D ARRAY CONTAINING THE AMPLITUDES FOR THE TIME-VARYING
C                 STRESS-TIME HISTORIES
C SX()          1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAX STRESS
C                 COMPONENT
C SXY()         1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAXY STRESS
C                 COMPONENT
C SXZ()         1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAXZ STRESS
C                 COMPONENT
C SXST          STATIC SIGMAX STRESS COMPONENT
C SXYST         STATIC SIGMAXY STRESS COMPONENT
C SXZST         STATIC SIGMAXZ STRESS COMPONENT
C SY()          1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAY STRESS
C                 COMPONENT
C SYZ()         1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAYZ STRESS
C                 COMPONENT
C SZ()          1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAZ STRESS
C                 COMPONENT
C SYST          STATIC SIGMAY STRESS COMPONENT
C SYZST         STATIC SIGMAYZ STRESS COMPONENT
C SZST          STATIC SIGMAZ STRESS COMPONENT
C T()           1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C TEST          UNIFORM(0,1) RANDOM VARIATE USED TO DETERMINE
C                 HYPERDISTRIBUTION TO SELECT FROM
C THIC          SELECTED WALL THICKNESS AT WELD, IN \
C THICA         WALL THICKNESS  LOWER BOUND          |
C THICB         WALL THICKNESS  UPPER BOUND          |
C THICR         SELECTED RHO FOR WALL THICKNESS      |  BETA HYPER -
C THICR1        WALL THICKNESS  - RHO LOWER BOUND     \ DISTRIBUTION
C THICR2        WALL THICKNESS  - RHO UPPER BOUND     / PARAMETERS
C THICT         SELECTED THETA FOR WALL THICKNESS    |
C THICT1        WALL THICKNESS  - THETA LOWER BOUND  |
C THICT2        WALL THICKNESS  - THETA UPPER BOUND /
C TIN           SELECTED INNER WALL SURFACE TEMPERATURE (RANKINE)     \
C TIMU          SELECTED MEAN OF INNER WALL TEMPERATURE                |
C TIMUA         MEAN OF INNER WALL TEMPERATURE LOWER BOUND             | NORMAL
C TIMUB         MEAN OF INNER WALL TEMPERATURE UPPER BOUND             \ HYPER-
C TISIG         SELECTED STD DEVIATION OF INNER WALL TEMPERATURE,      / DISTR
C TISIGA        STD DEVIATION OF INNER WALL TEMPERATURE LOWER BOUND   |
C TISIGB        STD DEVIATION OF INNER WALL TEMPERATURE UPPER BOUND  /
C TOUT          SELECTED OUTER WALL SURFACE TEMPERATURE (RANKINE)     \
C TOMU          SELECTED MEAN OF OUTER WALL TEMPERATURE                |
C TOMUA         MEAN OF OUTER WALL TEMPERATURE LOWER BOUND             | NORMAL
C TOMUB         MEAN OF OUTER WALL TEMPERATURE UPPER BOUND             \ HYPER-
C TOSIG         SELECTED STD DEVIATION OF OUTER WALL TEMPERATURE       / DISTR
C TOSIGA        STD DEVIATION OF OUTER WALL TEMPERATURE LOWER BOUND   |
C TOSIGB        STD DEVIATION OF OUTER WALL TEMPERATURE UPPER BOUND  /
C TSTAT         STATIC TORQUE LOADS
C TRUNC         VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
C TYPE()        1-D ARRAY CONTAINING THE TYPE OF TIME-VARYING LOAD, USED FOR
C                 LOAD FACTORS -- TYPE(*) = 1 INDICATES NARROW-BAND RANDOM;
```

```
C                   TYPE(*) = 2 INDICATES SUPERIMPOSED SINUSOID; TYPE(*) = 3
C                   INDICATES AERODYNAMIC
C V()             2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C                   ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C VSTAT()         1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- VSTAT(1) IS
C                   THE V2 LOAD; VSTAT(2) IS THE V3 LOAD
C WIDTH           SELECTED PLATE WIDTH, IN  \
C WITHA           WIDTH LOWER BOUND          |
C WITHB           WIDTH UPPER BOUND          |
C WITHR           SELECTED RHO FOR WIDTH     |  BETA HYPER
C WITHR1          WIDTH - RHO LOWER BOUND     \ DISTRIBUTION
C WITHR2          WIDTH - RHO UPPER BOUND     / PARAMETERS
C WITHT           SELECTED THETA FOR WIDTH   |
C WITHT1          WIDTH - THETA LOWER BOUND  |
C WITHT2          WIDTH - THETA UPPER BOUND /
C WOFF            SELECTED WELD OFFSET (%)                               \
C WOFFA           WELD OFFSET LOWER BOUND - HYPERDISTRIBUTION 1           |
C WOFFB           WELD OFFSET UPPER BOUND - HYPERDISTRIBUTION 1           |
C WOFFC           WELD OFFSET LOWER BOUND - HYPERDISTRIBUTION 2           |
C WOFFD           WELD OFFSET UPPER BOUND - HYPERDISTRIBUTION 2           |
C WOFFE           PERCENTAGE OCCURING IN HYPERDISTRIBUTION 1              |
C WOFFHI          SELECTED WELD OFFSET UPPER BOUND                        |
C WOFFLO          SELECTED WELD OFFSET LOWER BOUND                        |
C WOFFR           SELECTED RHO FOR WELD OFFSET                             \ BETA
C WOFFR1          WELD OFFSET - RHO LOWER BOUND - HYPERDISTRIBUTION 1      / HYPER
C WOFFR2          WELD OFFSET - RHO UPPER BOUND - HYPERDISTRIBUTION 1     |  DISTR
C WOFFR3          WELD OFFSET - RHO LOWER BOUND - HYPERDISTRIBUTION 2     |
C WOFFR4          WELD OFFSET - RHO UPPER BOUND - HYPERDISTRIBUTION 2     |
C WOFFT           SELECTED THETA FOR WELD OFFSET                          |
C WOFFT1          WELD OFFSET - THETA LOWER BOUND - HYPERDISTRIBUTION 1   |
C WOFFT2          WELD OFFSET - THETA UPPER BOUND - HYPERDISTRIBUTION 1   |
C WOFFT3          WELD OFFSET - THETA LOWER BOUND - HYPERDISTRIBUTION 2   |
C WOFFT4          WELD OFFSET - THETA UPPER BOUND - HYPERDISTRIBUTION 2  /
C==============================================================================

C========== OPEN THE INPUT AND OUTPUT FILES ============================

      OPEN (1, FILE = 'CRKDAT', STATUS = 'OLD')
      OPEN (3, FILE = 'CRKRES', STATUS = 'NEW')
      OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
      OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')

C=======================================================================

C  SET DEFAULT VALUES

      CALL SETDEF (LIFE, NCRL)

C  READ AND ECHO GENERAL DATA

      CALL INPUT (ANGLE, BLFPER, COEXP, E, EM,
     &            LOCAT, NBLIFE, NHYPER, NLIFE, NLIFET, NRAN,
     &            NU, NUMSEG, PERIOD, RSO, SE, STRHIS, TRUNC)

C  READ MATERIAL PROPERTIES AND PERFORM REGRESSION ON CRACK GROWTH DATA

      CALL GRODAT (CEE, CO, DEE, DKTHO, EMM, ENN, FTY, KC, PEE, QUE)

C  FOR HEX COIL GROW CRACK IN TWO DIRECTIONS BUT FOR EXHEX ONLY ONE

      IF(KPROB .EQ. 1) THEN
         NDIR = 2
      ELSE
         NDIR = 1
      ENDIF

C >>>> THIS LOOP SAMPLES HYPERPARAMETER SETS <<<<

      DO 300 K = 1, NHYPER

         CALL HYPDRW (AERD, AERS, ASTR, DSTR, KLAM, LAMGR, LAMKC,
     &                LAMKH, LAMW, NEUB, SSTR, MVAR)

C IF COEFFICIENT m IS VARYING
```

```
         IF(KGROW .EQ. 2) THEN
            EMM = MVAR
         ENDIF

C >>>> THIS LOOP GENERATES CRACK GROWTH LIVES <<<<

         DO 200 I = 1, NLIFE

C PERFORM DRIVER DRAWS

            CALL PARDRW (AI, CI, DLTAT, INDIA, LAMN, LAMS, PC,
     &                   THIC, WIDTH, WOFF)

C PERFORM CRACK GROWTH LIFE CALCULATION

            CALL LIFCAL (AERD, AERS, ASTR, AI, ANGLE, CI, CEE,
     &             CO, COEXP, DEE, DKTHO, DLTAT, DSTR, E, EM,
     &             EMM, ENN, FTY, INDIA, KC, KLAM, LAMGR, LAMKC,
     &             LAMKH, LAMN, LAMS, LAMW, LOCAT, NEUB, NEWLIF,
     &             NDIR, NCRL, NRAN, NU, NUMSEG, PC, PEE, PERIOD,
     &             QUE, RSO, SE, SSTR, STRHIS, THIC, TRUNC, WIDTH, WOFF)

C SAVE AND SORT THE SHORTEST 1% OF LIVES AFTER SORTING

            IF (NLIFET .GT. 1) THEN
               CALL INSORT (NEWLIF, LIFE, NLIFET)
            ENDIF
  200    CONTINUE

  300 CONTINUE

C  WRITE OUT THE LIVES AND BLIVES

      WRITE(3,1000)
      IF (NLIFET .GT. 1) THEN
         NUM = NLIFET/100
         WRITE (3,1200)
         DO 400 I = 1, NUM
            WRITE(3,1100) LIFE(I)
            WRITE(9,*) I, FLOAT(I)/FLOAT(NLIFET), LIFE(I)
  400    CONTINUE
         WRITE(3,1300)
         DO 500 J = 1, NBLIFE
            BLFPOS = NINT (BLFPER(J) * FLOAT (NLIFET))
            WRITE(3,1400) BLFPER(J), LIFE(BLFPOS)
  500    CONTINUE
      ELSE
         WRITE(3,1500) NEWLIF
      ENDIF

      STOP
C------------------------ FORMATS ---------------------------------
 1000 FORMAT(///,30X,'SIMULATION OUTPUT',///)
 1100 FORMAT(20X,E12.5)
 1200 FORMAT(13X,'SHORTEST 1% OF CRACK GROWTH LIVES ',//,
     &       20X,'    LIFE ',/)
 1300 FORMAT(///,2X,'B LIVES:       EMPIRICAL',/)
 1400 FORMAT(2X,F7.5,5X,E12.5)
 1500 FORMAT(13X,'CRACK GROWTH LIFE = ',E12.5)
      END

C***********************************************************************
C SUBROUTINE LIFCAL CALCULATES CRACK GROWTH LIFE
C
C PROGRAMMER: S. SUTHARSHANA
C DATE : DECEMBER 1992
C VERSION : 92.5
C
C  Copyright (C) 1991, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************
      SUBROUTINE LIFCAL (AERD, AERS, ASTR, AI, ANGLE, CI, CEE,
     &             CO, COEXP, DEE, DKTHO, DLTAT, DSTR, E, EM,
```

```
     &              EMM, ENN, FTY, INDIA, KC, KLAM, LAMGR, LAMKC,
     &              LAMKH, LAMN, LAMS, LAMW, LOCAT, NEUB, NEWLIF,
     &              NDIR, NCRL, NRAN, NU, NUMSEG, PC, PEE, PERIOD,
     &              QUE, RSO, SE, SSTR, STRHIS, THIC, TRUNC, WIDTH, WOFF)

C  SUBPROGRAMS: STRAN1, STRAN2, CYCOUN, BLKGRO

C      IMPLICIT NONE

      INTEGER  J, JLAST, NBIN(100), MAXLD, MAXM, MAXSEG

      PARAMETER (MAXLD = 16, MAXM = 20000, MAXSEG = 10)

C=================== LOCAL VARIABLES ===========================

      REAL     A(2), AF, AOC, DADB(2), DELA, DELC, DSALT,
     &         NEWA(101), PDADB, PDCDB, RATIO, SM, SPR(MAXM), TOTLIF
      LOGICAL FAIL

C A()      CRACK LENGTH IN THE "a" AND "c" DIRECTIONS
C DELA     CRACK LENGTH INCREMENT IN THE "a" DIRECTION
C DELC     CRACK LENGTH INCREMENT IN THE "c" DIRECTION
C NEWA()   ARRAY OF CRACK LENGTHS TO PERFORM BLOCK GROWTH CALCULATIONS AT
C PDADB    PREVIOUS da/dB

C================== EXTERNAL VARIABLES INPUT AND OUTPUT =============

      INTEGER INEUB, IRET, KGROW, KPROB

      INTEGER  IOUT, LOCAT, NDIR, NCRL, NLOAD, NRAN,
     &         NUMSEG, TYPE(MAXLD)

      REAL    AERD, AERS, ASTR, AI, ANGLE, CI, CEE, CO,
     &        COEXP, DEE, DKTHO, DLTAT, DSTR, E(MAXSEG), EM,
     &        EMM, ENN, FTY, INDIA, KC, KLAM,
     &        LAMGR, LAMKC, LAMKH, LAMN, LAMS, LAMW, M(2,MAXLD),
     &        MSTAT(2), NEUB, NEWLIF, NU, P(MAXLD), PC, PCO, PEE,
     &        PERIOD, PSTAT, QUE, RSO, SE(MAXSEG), SSTR,
     &        STRHIS(MAXLD,MAXM),
     &        SX(MAXLD), SXST, SXY(MAXLD), SXYST, SXZ(MAXLD),
     &        SXZST, SY(MAXLD), SYST, SYZ(MAXLD), SYZST, SZ(MAXLD), SZST,
     &        T(MAXLD), THIC, TRUNC, TSTAT, V(2,MAXLD), VSTAT(2),
     &        WIDTH, WOFF

      CHARACTER*6 LDNAME(MAXLD)

      COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
      COMMON/NAMES/LDNAME
      COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
     &             P, T, M, V, PCO,
     &             SXST, SYST, SZST, SXYST, SXZST, SYZST,
     &             SX, SY, SZ, SXY, SXZ, SYZ
      COMMON IOUT

C  PERFORM LOAD TO STRESS TRANSFORMATION

      IF (KPROB .EQ. 1) THEN
         AF = THIC
         CALL STRAN1 (AERD, AERS, ASTR, ANGLE, COEXP, DLTAT, DSTR, EM,
     &                INDIA, LAMN, LAMS, LAMW, LOCAT, NRAN, NU, PC,
     &                SPR, STRHIS, THIC, WOFF)
      ELSEIF (KPROB .EQ. 2) THEN
         AF = WIDTH/2.0
         CALL STRAN2 (DSTR, LAMN, LAMS, NRAN, SPR, SSTR, STRHIS)
      ENDIF

C  PERFORM CYCLE COUNTING

      CALL CYCOUN (DSALT, E, EM, NBIN, NEUB, NUMSEG, NRAN, SE,
     &             SPR, SM, TRUNC)

C  ESTABLISH CRACK LENGTHS AT WHICH BLOCK GROWTH CALCULATIONS ARE PERFORMED

      NEWA(1) = AI
      DELA = EXP( LOG(AF/AI)/FLOAT(NCRL) )
```

```
      DO 50 J = 1, NCRL
         NEWA(J+1) = NEWA(J)*DELA
 50   CONTINUE

      A(1) = NEWA(1)
      TOTLIF = 0.0
      FAIL = .FALSE.
      JLAST = 1
      IF(NDIR .EQ. 2) THEN
         A(2) = CI
      ENDIF
      PDADB = 0.0
      PDCDB = 0.0

C  CALCULATE CRACK-GROWTH LIFE FOR THE LOAD BLOCK

C >>>> THIS LOOP IS FOR EVERY CRACK LENGTH

      DO 100 J = 1, NCRL

         DADB(1) = 0.0
         DADB(2) = 0.0

         CALL BLKGRO (A, CEE, CO, DADB, DEE, DKTHO,
     &                DSALT, EMM, ENN, FAIL, FTY, INDIA, KC, KLAM,
     &                LAMKC, LAMKH, NBIN, NDIR, PEE,
     &                QUE, RSO, SM, THIC, WIDTH)

         IF(IOUT .EQ. 20) THEN
            WRITE(8,*) A(1), A(2), DADB(1), DADB(2)
            IF(NDIR .EQ. 2) THEN
               AOC = A(1)/A(2)
               WRITE(8,*) AOC
            ENDIF
         ENDIF

         IF(PDADB .GT. 0.0) THEN
            DELA = NEWA(J) - NEWA(J-1)
            TOTLIF = 2.0*DELA/(DADB(1) + PDADB) + TOTLIF
         ELSEIF(PDCDB .GT. 0.0) THEN
            TOTLIF = 2.0*DELC/(DADB(2) + PDCDB) + TOTLIF
         ENDIF

         IF (DADB(1) .GT. 0.0) THEN
            A(1) = NEWA(J+1)
            IF(NDIR .EQ. 2) THEN
               RATIO = DADB(2)/DADB(1)
               DELC = (NEWA(J+1) - NEWA(J)) * RATIO
               A(2) = A(2) + DELC
            ENDIF
            IF(FAIL) THEN
               FAIL = .FALSE.
               WRITE(8,*)'K GT Kcr AT A = ',A(1)
               GO TO 110
            ENDIF
            JLAST = J
         ELSE
            IF(FAIL) THEN
               FAIL = .FALSE.
               WRITE(8,*)'K GT Kcr AT A = ',A(1)
               GO TO 110
            ELSE
               IF(NDIR .EQ. 1 .OR. A(2) .GT. WIDTH/2.0
     &            .OR. DADB(2) .EQ. 0.0) THEN
                  TOTLIF = 1.0E+37
                  WRITE(8,*)'NO GROWTH AT', J,'th CRACK LENGTH'
                  GO TO 110
               ENDIF

               IF(NDIR .EQ. 2 .AND. A(2) .LT. WIDTH/2.0) THEN
                  DELC = A(2)*( EXP(LOG(WIDTH/(2.0*A(2)))/
     &                   FLOAT(NCRL-J+1)) - 1.0)
                  A(2) = A(2) + DELC
                  A(1) = NEWA(JLAST)
               ENDIF
```

```
            ENDIF
            WRITE(8,*) 'NO GROWTH IN A DIRECTION AT, J,'th CRACK LENGTH'
         ENDIF
         PDADB = DADB(1)
         PDCDB = DADB(2)
CC          WRITE(8,*) A(1), A(2), DELA, DELC, TOTLIF
  100 CONTINUE

C  CALCULATE LIFE

  110 CONTINUE
      NEWLIF =  LAMGR * PERIOD * TOTLIF

      RETURN
      END
C***********************************************************************
C SUBROUTINE BLKGRO CALCULATES THE CRACK GROWTH RATE PER BLOCK
C
C PROGRAMMER : S. SUTHARSHANA
C
C DATE : DECEMBER 1992
C VERSION: 92.5
C
C  Copyright (C) 1991, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      SUBROUTINE BLKGRO (A, CEE, CO, DADB, DEE, DKTHO, DSALT,
     &                   EMM, ENN, FAIL, FTY, INDIA, KC, KLAM,
     &                   LAMKC, LAMKH, NBIN, NDIR, PEE, QUE, RSO, SM,
     &                   THIC, WIDTH)

C  SUBPROGRAMS: STRIF1, STRIF2

C       IMPLICIT NONE

      INTEGER I, IDIR, IOUT, NBIN(100), NDIR

      INTEGER INEUB, IRET, KGROW, KPROB

      REAL A(2), AB(2), AO(2), AORPO, AORPA, ARPI, CEE,
     &     CO, CONST, DA, DADB(2), DEE, DK,
     &     DKEFF, DKTH, DKTHO, DSALT, EMM, ENN,
     &     F0(2), F2(2), FTY, INDIA, KC, KCR, KLAM,
     &     KMAX(2), KMAXEF, KMAXRQ, KMIN(2), KMINEF, LAMKC, LAMKH,
     &     PI, PLSR(2), PEE, QUE, REFF, RPI,
     &     RPO(2), RSO, SALMAX, SALTF, SM, THIC, WIDTH

C====================== DESCRIPTION OF LOCAL VARIABLES ==================
C  AB()     CRACK LENGTHS DURING GROWTH IN THE BLOCK
C  AO()     CRACK LENGTHS AT THE LAST OVERLOAD
C  AORPO    AO + RPO
C  AORPA    AORPO - AB
C  ARPI     AORPO - AB
C  F0()     SIF COEFF FOR TENSILE STRESS
C  F2()     SIF COEFF FOR BENDING STRESS
C  KMAXRQ   REQUIRED SIF FOR WILLENBORG MODEL
C  PLSR()   PLANE STRAIN/STRESS PLASTIC ZONE SIZE COEFF
C  RPI      CURRENT PLASTIC ZONE SIZE
C  RPO()    OVERLOAD PLASTIC ZONE SIZE
C=======================================================================
      LOGICAL FAIL

      COMMON/CNTRL/INEUB, IRET, KGROW, KPROB

      COMMON IOUT

      DATA PI/3.14159265358979/, PLSR/0.053051647, 0.159154943/

      AB(1) = A(1)
      AB(2) = A(2)
      AO(1) = 0.0
      AO(2) = 0.0
      RPO(1) = 0.0
```

```
      RPO(2) = 0.0

      IF(IOUT .EQ. 20) THEN
         WRITE(8,*) 'INSIDE BLKGRO ROUTINE'
         WRITE(8,*) 'A, C ',A(1), A(2)
      ENDIF

C  CALCULATE THE STRESS INTENSITY FACTOR COEFFICIENTS

      CONST = SQRT(PI*A(1)) * KLAM
      IF(KPROB .EQ. 1) THEN
         CALL STRIF1(A(1), A(2), F0, F2, INDIA, THIC)
      ELSEIF(KPROB .EQ. 2) THEN
         CALL STRIF2(A(1), F0, F2, WIDTH)
      ENDIF

C  LOOP FOR EVERY STRESS CYCLE IN HISTORY
C  AND LOOP FOR 'a' DIRECTION (=1) AND 'c' DIRECTION (=2)

      SALMAX = DSALT*101.0

      DO 200 I = 1, 100

         IF (NBIN(I) .GT. 0) THEN

         SALTF = SALMAX - FLOAT(I)*DSALT

         DO 100 IDIR=1,NDIR

            KMAX(IDIR)  = CONST*(F0(IDIR)*SALTF + F2(IDIR)*SM)
            KMIN(IDIR)  = CONST*(-F0(IDIR)*SALTF + F2(IDIR)*SM)

            IF(IOUT .EQ. 20) THEN
               WRITE(8,*) 'DIRECTION = ', IDIR
               WRITE(8,*) 'DIR, KMAX 1,2, KMIN 1,2', IDIR, KMAX(IDIR),
     &         KMIN(IDIR)
            ENDIF

C  IF MAXIMUM SIF IS NEGATIVE OR ZERO NO GROWTH IN THIS DIRECTION

            IF(KMAX(IDIR).LE.0.0) THEN
               GO TO 95
            ENDIF

C           RESET MINIMUM SIF TO ZERO IF NEGATIVE

            IF(KMIN(IDIR).LE.0.0) THEN
               KMIN(IDIR) = 0.0
            ENDIF

            DKEFF = KMAX(IDIR) - KMIN(IDIR)
            REFF = KMIN(IDIR)/KMAX(IDIR)
            KMAXEF = KMAX(IDIR)

            IF (IRET .EQ. 1 ) THEN
               AORPO = AO(IDIR) + RPO(IDIR)
               RPI = PLSR(IDIR) * (KMAX(IDIR)/FTY)**2
               ARPI = AB(IDIR) + RPI
               IF(ARPI .GT. AORPO) THEN
                  RPO(IDIR) = RPI
                  AO(IDIR) = AB(IDIR)
               ELSE
                  AORPA = AORPO - AB(IDIR)
                  KMAXRQ = FTY*( AORPA/PLSR(IDIR))**(0.5)
                  KMAXEF = KMAX(IDIR) - (KMAXRQ - KMAX(IDIR))/(RSO-1.0)
                  KMINEF = KMIN(IDIR) - (KMAXRQ - KMAX(IDIR))/(RSO-1.0)
                  IF( KMAXEF.GT.0.0 ) THEN
                     IF( KMINEF.GT.0.0 ) THEN
                        DKEFF = KMAXEF - KMINEF
                        REFF = KMINEF/KMAXEF
                     ELSE
                        DKEFF = KMAXEF
                        REFF = 0.0
                     ENDIF
                  ELSE
```

```
                  GO TO 95
               ENDIF
            ENDIF
            IF(IOUT .EQ. 20) THEN
               WRITE(8,*) 'AORPA, AORPO, KMAXRQ, RPI, RPO, AO',
     &                     AORPA, AORPO, KMAXRQ, RPI, RPO(IDIR),
     &                     AO(IDIR)
            ENDIF
         ENDIF

C        CALCULATE BLOCK CRACK GROWTH RATE AND NEW CRACK LENGTH

         IF(IOUT .EQ. 20) THEN
            WRITE(8,*) 'CYC, DIR, DKEFF, REFF ',I,IDIR,DKEFF,REFF
         ENDIF

C        CHANGE FROM PSI TO KSI

         DK = DKEFF/1000.0

C        CALCULATE CRACK GROWTH

         IF( REFF .LE. 0.9 ) THEN
            DKTH=LAMKH*DKTHO*(1.0 - CO*REFF)**DEE
         ELSE
            DKTH=LAMKH*(DKTHO*(1.0 - 0.90*CO)**DEE)
     &               *10.0*(1.0 - REFF)
         ENDIF

         KCR = KC * LAMKC
         KMAXEF = KMAXEF/1000.0

         IF ( (DK. GT. DKTH) .AND. (KMAXEF .LE. KCR) ) THEN
            DA = FLOAT(NBIN(I)) *
     &           CEE * ((1. - REFF)**EMM) * (DK)**ENN
     &           * (DK - DKTH)**PEE
     &         / ( ( (1. - REFF) * KCR - DK )**QUE )
            DADB(IDIR) = DADB(IDIR) + DA
            AB(IDIR) = AB(IDIR) + DA
            IF(IOUT .EQ. 20) THEN
               WRITE(8,*)'DIR, DA, DADB, AB ',IDIR, DA,
     &                   DADB(IDIR),AB(IDIR)
            ENDIF
         ELSEIF (KMAXEF .GT. KCR ) THEN
            FAIL = .TRUE.
            RETURN
         ENDIF

 95      CONTINUE
100   CONTINUE
      ENDIF
200 CONTINUE

    RETURN
    END

C==================================================================
C  SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  12MAY88
C     VERSION:  2
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

    SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

C  INPUTS:   NEWLIF, LIFE, NLIFET
C  OUTPUTS:  LIFE

C      IMPLICIT NONE

    COMMON   IOUT
```

```
      INTEGER  I, IOUT, MAXLIF, NLIFET, NUM, PLACE
      PARAMETER (MAXLIF = 1000)
      REAL     LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

C                    LIST OF VARIABLES
C
C  I          CONTROLS DO LOOP FOR INSERTION
C  IOUT       OUTPUT DUMP CONTROLLER
C  LIFE()     1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
C               PFM TO BE SORTED
C  MAXLIF     MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
C               CALCULATION
C  NEWLIF     LIFE VALUE TO BE INSERTED INTO LIFE()
C  NLIFET     TOTAL NUMBER OF LIVES CALCULATED BY PFM
C  NUM        NUMBER OF LIFE VALUES IN LIFE()
C  PLACE      POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
C  TEMP()     1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON INSERTION
C               OF NEWLIF

      NUM = NLIFET / 100

C     FIND POSITION IN LIFE() FOR NEWLIF

      IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

      DO 100 I = 1, NUM
         IF (NEWLIF .LT. LIFE(I)) THEN
            PLACE = I
            GOTO 110
         ENDIF
  100 CONTINUE
  110 CONTINUE

C     STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()

      DO 200 I = (PLACE + 1), NUM
         TEMP(I) = LIFE(I-1)
  200 CONTINUE

C     INSERT NEWLIF

      LIFE(PLACE) = NEWLIF

C     SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

      DO 300 I = (PLACE + 1), NUM
         LIFE(I) = TEMP(I)
  300 CONTINUE

C     IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN

  400 CONTINUE


      RETURN
      END
C***********************************************************************
C   SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C   INDEPENDENT RANDOM VARIATES
C   PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C         DATE:  9MAR87
C   SUBPROGRAM:  RANDOM
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

      COMMON  IOUT
```

```
      DOUBLE PRECISION RAND

      REAL    FRAC, RHO1, RHO2, THE1, THE2, X, Y

      INTEGER IOUT

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      X = FRAC * (RHO2 - RHO1) + RHO1

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      Y = FRAC * (THE2 - THE1) + THE1

      IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ' RHO2 =', RHO2,
     &        ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y

      RETURN
      END
C*********************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C*********************************************************************

      SUBROUTINE RANDOM (FRAC, RAND)
C      IMPLICIT NONE
      COMMON  IOUT
      INTEGER IOUT
      REAL    FRAC
      DOUBLE PRECISION  RANA, RANC, RAND, RANDIV, RANM, RANSUB,
     &        RANT, RANX


C                LIST OF VARIABLES
C
C   FRAC       UNIFORM (0,1) RANDOM VARIATE
C   IOUT       OUTPUT DUMP CONTROLLER
C   RANA       CONSTANT FOR LCG
C   RANC       CONSTANT FOR LCG
C   RAND       RANDOM NUMBER SEED
C   RANDIV     INTERNAL CALCULATION
C   RANM       CONSTANT FOR LCG
C   RANSUB     INTERNAL CALCULATION
C   RANT       INTERNAL CALCULATION
C   RANX       INTERNAL CALCULATION


C          USING LCG RANDOM # GENERATOR

      RANA = 671093.0
      RANC = 7090885.0
      RANM = 33554432.0

   10 RANX = RANA * RAND + RANC
      RANDIV = RANX / RANM
      RANT = DINT(RANDIV)
      RANSUB = RANT * RANM
      RAND = RANX - RANSUB
      FRAC = SNGL(RAND / RANM)

      IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
```

```
      IF (IOUT .EQ. 2) WRITE(8,*)'RANX =', RANX, ' RANDIV =', RANDIV,
     &    ' RANT =', RANT, ' RANSUB =', RANSUB, ' RAND =', RAND,
     &    ' FRAC =', FRAC

      RETURN
      END

C     NOTES:  IOUT=2 DUMPS TO SCREEN

C**********************************************************************
C   SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C   WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C   PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C         DATE:  3FEB88
C      VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
C
C   The random variates are generated using the "Direct Method"
C   Abramowitz, M., and Stegun, I. A., editors, Handbook of
C     Mathematical Functions, National Bureau of Standards, Applied
C     Mathematics Series 55, Issued June 1964, Ninth Printing, November
C     1970 with corrections, pg. 953.
C**********************************************************************

      SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C     SUBPROGRAM:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL     FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2

      PARAMETER (PI = 3.1415926536)

      INTEGER IOUT

C
C              LIST OF VARIABLES
C
C   FRAC     UNIFORM(0,1) RANDOM VARIATE
C   IOUT     OUTPUT DUMP CONTROLLER
C   MU       MEAN OF NORMAL DISTRIBUTION
C   RAND     RANDOM NUMBER SEED
C   SIGMA    STANDARD DEVIATION OF NORMAL DISTRIBUTION
C   X        NORMAL RANDOM VARIATE
C   U1       UNIFORM RANDOM NUMBER U(0,1)
C   U2       UNIFORM RANDOM NUMBER U(0,1)
C   Z1       NORMAL RANDOM NUMBER ON N(0,1)
C   Z2       NORMAL RANDOM NUMBER ON N(0,1)

      IF (IOUT .EQ. 15)
     &   WRITE(8,*) 'RAND =', RAND, ' MU =', MU, ' SIGMA =', SIGMA

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

      CALL RANDOM (FRAC, RAND)
      U2 = FRAC
      IF (IOUT .EQ. 15)
     &   WRITE(8,*) 'U1 =', U1,' U2 =', U2

      Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
      Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)

      X = SIGMA * Z1 + MU
      IF (IOUT .EQ. 15)
     &   WRITE(8,*) 'Z1 =', Z1, ' Z2 =', Z2, ' X =', X

      RETURN
      END
```

```
C*************************************************************************
C  SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C  ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  5OCT87
C*************************************************************************

      SUBROUTINE TRMNAT

      WRITE(8,*) 'PROGRAM EXECUTION TERMINATED'
      STOP
      END


C*************************************************************************
C  SUBROUTINE M4L1 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
C  FOR LOCATION 1 (PLAIN WELD, EXTERIOR SURFACE OF THE DUCT) UNDER THERMAL
C  LOADING
C  PROGRAMMER:  L. NEWLIN
C        DATE:  JUL92
C     VERSION:  92.4
C*************************************************************************

      SUBROUTINE M4L1 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT,
     &                 NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP,
     &                 T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT)

C  INPUTS:  ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT, NLOAD, NU, P,
C           PC, PCO, PSTAT, STATIC, T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT
C
C  OUTPUTS:  STATIC, STRAMP

C     IMPLICIT NONE

      COMMON   IOUT

      INTEGER  I, IOUT, J, MAXLD, NLOAD

      REAL     PI

      PARAMETER (MAXLD = 16, PI = 3.1415926536)

      REAL     ALPHA, ANGLE, AREA, DI, DLTAT, EM, FK(10), GEOM, IFK,
     &         K(2, 2), KOFF, LAMW, M(2, MAXLD), MI, MSTAT(2), NU,
     &         P(MAXLD), PC, PCO, PSTAT, RDIFF, RI, RI2, RO, RO2,
     &         ROT, RT(10), SIG1A(MAXLD), SIG1B(MAXLD), SKT1, SKT2,
     &         STATIC(4), STHMA, STR1A, STR2A, STR1B, STR2B, STR1C,
     &         STRAMP(4, MAXLD), T(MAXLD), THIC, TSTAT, V(2, MAXLD),
     &         VSTAT(2), WOFF


C                  LIST OF VARIABLES
C
C  ALPHA     COEFFICIENT OF THERMAL EXPANSION
C  ANGLE     ANGLE THETA IN RADIANS
C  AREA      CROSS SECTION AREA OF DUCT WALL
C  DI        INTERIOR DIAMETER
C  DLTAT     TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES
C  EM        YOUNG'S MODULUS PRIOR TO YIELD
C  FK()      1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C              CONCENTRATION DUE TO WELD ECCENTRICITY
C  GEOM      INTERMEDIATE THERMAL STRESS CALCULATION VARIABLE
C  I         CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND
C              AERODYNAMIC LOADS
C  IFK       INTERPOLATED VALUE OF Fk CORRESPONDING TO THE VALUE OF r/t
C  IOUT      OUTPUT DUMP CONTROLLER
C  J         CONTROLS DO LOOP FOR EACH POINT IN RT() AND FK() DURING
C              INTERPOLATION
C  K()       FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR DUCT
C              EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR DUCT EXTERIOR
C              FOR HOOP DIRECTION; K(1,2) IS FOR DUCT INTERIOR FOR AXIAL
```

```
C                    DIRECTION; K(2,2) IS FOR DUCT INTERIOR FOR HOOP DIRECTION
C   KOFF           STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
C   LAMW           ACCURACY FACTOR OF Fk - r/t CURVE
C   M()            2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                    ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C   MAXLD          MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C   MI             MOMENT OF INERTIA
C   MSTAT()        1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE
C                    M2 LOAD; M(2) IS THE M3 LOAD
C   NLOAD          NUMBER OF TIME-VARYING LOADS
C   NU             POISSON'S RATIO
C   P()            1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C   PC             LIMIT PRESSURE ON INSIDE OF THE VESSEL
C   PCO            LIMIT PRESSURE ON OUTSIDE OF THE VESSEL
C   PI             SELF EXPLANATORY CONSTANT
C   PSTAT          STATIC AXIAL LOAD
C   RDIFF          EQUAL TO RO2 - RI2
C   RI             INTERIOR RADIUS
C   RI2            INNER RADIUS SQUARED
C   RO             OUTER RADIUS
C   RO2            OUTER RADIUS SQUARED
C   ROT            EQUAL TO r / t (R Over T)
C   RT()           1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C                    CONCENTRATION DUE TO WELD ECCENTRICITY
C   SIG1A()        1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO FORCE
C                    FOR THE TIME-VARYING LOADS
C   SIG1B()        1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO BENDING
C                    FOR THE TIME-VARYING LOADS
C   SKT1           STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
C   SKT2           STRESS CONCENTRATION FACTOR FOR HOOP STRESS
C   STATIC()       1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES --
C                    STATIC(1) IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS;
C                    STATIC(3) IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C   STHMA          THE STATIC AXIAL STRESS DUE TO THERMAL GRADIENT
C   STR1A          THE STATIC AXIAL STRESS DUE TO FORCE
C   STR1B          THE STATIC AXIAL STRESS DUE TO BENDING
C   STR1C          THE STATIC AXIAL STRESS DUE TO MOMENTUM CHANGE (FLUID)
C   STR2A          THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO INTERNAL PRESSURE
C   STR2B          THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO EXTERNAL PRESSURE
C   STRAMP()       2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C                    -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
C                    THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C                    STRAMP(4,*) ARE THE SHEAR STRESSES
C   T()            1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C   THIC           WALL THICKNESS AT DUCT OUTER RADIUS
C   TSTAT          STATIC TORQUE LOAD
C   V()            2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C                    ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C   VSTAT()        1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V2
C                    LOAD; V(2) IS THE V3 LOAD
C   WOFF           WELD OFFSET

C     CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO
C     ECCENTRICITY OF THE WELD

      RI = DI / 2.0
      ROT = (DI + THIC) / (2.0 * THIC)


      DO 50 J = 2, 10
C        INTERPOLATE TO FIND FACTOR Fk CORRESPONDING TO VALUE OF r/t
         IF ((ROT .LE. RT(J)) .AND. (ROT .GE. RT(J-1))) THEN
            IFK = (FK(J) - FK(J-1)) * (ROT - RT(J-1))
     &            / (RT(J) - RT(J-1)) + FK(J-1)
         ENDIF
   50 CONTINUE


      KOFF = LAMW * (1.0 + 3.0 * IFK * WOFF)

      IF (IOUT .EQ. 25) THEN
         WRITE(8,*) 'DI = ', DI, ' RI = ', RI
         WRITE(8,*) 'THIC = ', THIC, ' ROT = ', ROT
         WRITE(8,*) 'IFK = ', IFK, ' WOFF = ', WOFF
```

```
            WRITE(8,*) 'LAMW = ', LAMW, ' KOFF = ', KOFF
         ENDIF

C        CALCULATE THE CROSS-SECTIONAL AREA AND MOMENT OF INERTIA

         AREA = PI * ((RI + THIC) ** 2 - RI ** 2)
         MI = PI * ((RI + THIC) ** 4 - RI ** 4) / 4.0

C        OBTAIN STRESS CONCENTRATION FACTORS AND RADII APPROPRIATE TO LOCATION
C        THIS IS THE EXTERIOR SURFACE

         SKT1 = K(1,1)
         SKT2 = K(2,1)
         RO = RI + THIC

         IF (IOUT .EQ. 25) THEN
            WRITE(8,*) 'AREA = ', AREA, ' MI = ', MI
            WRITE(8,*) 'K(1,1) = ', K(1,1), ' SKT1 = ', SKT1
            WRITE(8,*) 'K(2,1) = ', K(2,1), ' SKT2 = ', SKT2
            WRITE(8,*) 'THIC = ', THIC, ' RO = ', RO
            WRITE(8,*) 'ALPHA = ', ALPHA, ' NU = ', NU
            WRITE(8,*) 'DLTAT = ', DLTAT, ' EM = ', EM
            WRITE(8,*)
         ENDIF

         RI2 = RI ** 2
         RO2 = RO ** 2
         RDIFF = RO2 - RI2

         GEOM = 1.00 - 2.00 * LOG (RO / RI) * RI2 / RDIFF

C        TEMPERATURE STRESS

         STHMA = ((EM * ALPHA * DLTAT) / (2.00 * (1.00 - NU)
        &         * LOG (RO / RI))) * GEOM

C        AXIAL STRESS CALCULATIONS

         STR1A = PSTAT / AREA
         STR1B = (MSTAT(1) * COS (ANGLE) + MSTAT(2) * SIN (ANGLE)) * RO
        &         / MI
         STR1C = (PC - PCO) * RI2 / RDIFF

         STATIC(1) = (STR1A + STR1B + STR1C) * SKT1 * KOFF + STHMA

C        HOOP (2) AND RADIAL (3) STRESS CALCULATIONS

         STR2A = 2.0 * PC * RI2 / RDIFF
         STR2B = - PCO * (RO2 + RI2) / RDIFF

         STATIC(2) = (STR2A + STR2B) * SKT2 + STHMA

         STATIC(3) = - PCO

C        SHEAR STRESS

         STATIC(4) = TSTAT * RO / (2.0 * MI) - (2.0 / AREA
        &            * (VSTAT(1) * COS (ANGLE) + VSTAT(2) * SIN (ANGLE)))

         IF (IOUT.EQ.25) THEN
            WRITE(8,*) 'RO2 = ', RO2, ' RI2 = ', RI2
            WRITE(8,*) 'RDIFF = ', RDIFF, ' GEOM = ', GEOM
            WRITE(8,*) 'STATIC STRESS VALUES '
            WRITE(8,*) 'AXIAL STRESSES'
            WRITE(8,*) '    STR1A = ', STR1A, ' STR1B = ', STR1B
            WRITE(8,*) '    STR1C = ', STR1C, ' STHMA = ', STHMA
            WRITE(8,*) '    STATIC(1) = ', STATIC(1)
            WRITE(8,*) 'HOOP STRESSES'
            WRITE(8,*) '    STR2A = ', STR2A, ' STR2B = ', STR2B,
        &              ' STHMA = ', STHMA
            WRITE(8,*) '    STATIC(2) = ', STATIC(2)
            WRITE(8,*) 'RADIAL STRESS', '   STATIC(3) = ', STATIC(3)
            WRITE(8,*) 'SHEAR STRESS', '   STATIC(4) = ', STATIC(4)
```

```
            WRITE(8,*)
            ENDIF

         DO 100 I = 1, NLOAD

C        AXIAL STRESS CALCULATIONS

            SIG1A(I) = P(I) / AREA
            SIG1B(I) = (M(1,I) * COS (ANGLE) + M(2,I) * SIN (ANGLE))
     &                 * RO / MI

            STRAMP(1,I) = (SIG1A(I) + SIG1B(I)) * SKT1 * KOFF

C           HOOP (2) AND RADIAL (3) STRESSES ARE ZERO
C           BECAUSE PRESSURES ARE CONSTANT

            STRAMP(2,I) = 0.0

            STRAMP(3,I) = 0.0

C           SHEAR STRESS

            STRAMP(4,I) = T(I) * RO / (2.0 * MI) - (2.0 / AREA
     &                    * (V(1,I) * COS (ANGLE) + V(2,I) * SIN (ANGLE)))


            IF (IOUT.EQ.25) THEN
               WRITE(8,*) 'STRESS VALUES FOR I = ',I
               WRITE(8,*) 'AXIAL STRESSES'
               WRITE(8,*) '    SIG1A = ', SIG1A(I), ' SIG1B = ', SIG1B(I)
               WRITE(8,*) '    STRAMP(1,I) = ', STRAMP(1,I)
               WRITE(8,*) 'HOOP STRESSES', '    STRAMP(2,I) = ',STRAMP(2,I)
               WRITE(8,*) 'RADIAL STRESS', '    STRAMP(3,I) = ',STRAMP(3,I)
               WRITE(8,*) 'SHEAR STRESS', '   STRAMP(4,I) = ', STRAMP(4,I)
               WRITE(8,*)
            ENDIF

  100 CONTINUE


         IF (IOUT .EQ. 25) THEN
            WRITE(8,*) 'I      AXIAL      HOOP      RADIAL      SHEAR'
            WRITE(8,*) STATIC(1), STATIC(2), STATIC(3), STATIC(4)
            DO 300 I = 1, NLOAD
               WRITE(8,*) I, STRAMP(1,I), STRAMP(2,I), STRAMP(3,I),
     &                       STRAMP(4,I)
  300       CONTINUE
         ENDIF

         RETURN
         END


C***********************************************************************
C  SUBROUTINE M4L2 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
C  FOR LOCATION 2 (PLAIN WELD, INTERIOR SURFACE OF THE DUCT), UNDER
C  THERMAL LOADING
C  PROGRAMMER:  L. NEWLIN
C        DATE:  JUL92
C     VERSION:  92.4
C***********************************************************************

         SUBROUTINE M4L2 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT,
     &                    NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP,
     &                    T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT)

C  INPUTS:  ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT, NLOAD, NU,
C           P, PC, PCO, PSTAT, T, THIC, TSTAT V, VSTAT, WOFF, FK, RT
C
C  OUTPUTS:  STATIC, STRAMP
```

```
C       IMPLICIT NONE
        COMMON    IOUT
        INTEGER  I, IOUT, J, MAXLD, NLOAD
        REAL      PI
        PARAMETER (MAXLD = 16, PI = 3.1415926536)
        REAL      ALPHA, ANGLE, AREA, DLTAT, EM, FK(10), GEOM, IFK, DI,
     &            K(2, 2), KOFF, LAMW, M(2, MAXLD), MI, MSTAT(2), NU,
     &            P(MAXLD), PC, PCO, PSTAT, RDIFF, RI, RI2, RO, RO2,
     &            ROT, RT(10), SIG1A(MAXLD), SIG1B(MAXLD), SKT1, SKT2,
     &            STATIC(4), STHMA, STR1A, STR2A, STR1B, STR2B, STR1C,
     &            STRAMP(4, MAXLD), T(MAXLD), THIC, TSTAT, V(2, MAXLD),
     &            VSTAT(2), WOFF


C                        LIST OF VARIABLES
C
C   ALPHA        COEFFICIENT OF THERMAL EXPANSION
C   ANGLE        ANGLE THETA IN RADIANS
C   AREA         CROSS SECTION AREA OF DUCT WALL
C   DI           INTERIOR DIAMETER
C   DLTAT        TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES
C   EM           YOUNG'S MODULUS PRIOR TO YIELD
C   FK()         1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C                  CONCENTRATION DUE TO WELD ECCENTRICITY
C   GEOM         INTERMEDIATE THERMAL STRESS CALCULATION VARIABLE
C   I            CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND
C                  AERODYNAMIC LOADS
C   IFK          INTERPOLATED VALUE OF Fk CORRESPONDING TO THE VALUE OF r/t
C   IOUT         OUTPUT DUMP CONTROLLER
C   J            CONTROLS DO LOOP FOR EACH POINT IN RT() AND FK() DURING
C                  INTERPOLATION
C   K()          FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR DUCT
C                  EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR DUCT EXTERIOR
C                  FOR HOOP DIRECTION; K(1,2) IS FOR DUCT INTERIOR FOR AXIAL
C                  DIRECTION; K(2,2) IS FOR DUCT INTERIOR FOR HOOP DIRECTION
C   KOFF         STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
C   LAMW         ACCURACY FACTOR OF Fk - r/t CURVE
C   M()          2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                  ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C   MAXLD        MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C   MI           MOMENT OF INERTIA
C   MSTAT()      1-D ARRAY CONTAINING THE STATIC TIME-VARYING LOADS -- M(1) IS
C                  THE M2 LOAD; M(2) IS THE M3 LOAD
C   NLOAD        NUMBER OF TIME-VARYING LOADS
C   NU           POISSON'S RATIO
C   P()          1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C   PC           LIMIT PRESSURE ON INSIDE OF THE VESSEL
C   PCO          LIMIT PRESSURE ON OUTSIDE OF THE VESSEL
C   PI           SELF EXPLANATORY CONSTANT
C   PSTAT        STATIC AXIAL LOAD
C   RDIFF        EQUAL TO RO2 - RI2
C   RI           INTERIOR RADIUS
C   RI2          INNER RADIUS SQUARED
C   RO           OUTER RADIUS
C   RO2          OUTER RADIUS SQUARED
C   ROT          EQUAL TO r / t (R over T)
C   RT()         1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C                  CONCENTRATION DUE TO WELD ECCENTRICITY
C   SIG1A()      1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO FORCE
C                  FOR THE TIME-VARYING LOADS
C   SIG1B()      1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO BENDING
C                  FOR THE TIME-VARYING LOADS
C   SKT1         STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
C   SKT2         STRESS CONCENTRATION FACTOR FOR HOOP STRESS
C   STATIC()     1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES --
C                  STATIC(1) IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS;
C                  STATIC(3) IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C   STHMA        THE STATIC AXIAL STRESS DUE TO THERMAL GRADIENT
C   STR1A        THE STATIC AXIAL STRESS DUE TO FORCE
```

```
C  STR1B       THE STATIC AXIAL STRESS DUE TO BENDING
C  STR1C       THE STATIC AXIAL STRESS DUE TO MOMENTUM CHANGE (FLUID)
C  STR2A       THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO INTERNAL PRESSURE
C  STR2B       THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO EXTERNAL PRESSURE
C  STRAMP()    2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C                -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
C                THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C                STRAMP(4,*) ARE THE SHEAR STRESSES
C  T()         1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C  THIC        WALL THICKNESS AT DUCT OUTER RADIUS
C  TSTAT       STATIC TORQUE LOAD
C  V()         2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C                ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C  VSTAT()     1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V2
C                LOAD; V(2) IS THE V3 LOAD
C  WOFF        WELD OFFSET

C     CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO
C     ECCENTRICITY OF THE WELD

      RI = DI / 2.0
      ROT = (DI + THIC) / (2.0 * THIC)

      DO 50 J = 2, 10
C        INTERPOLATE TO FIND FACTOR Fk CORRESPONDING TO VALUE OF r/t
         IF ((ROT .LE. RT(J)) .AND. (ROT .GE. RT(J-1))) THEN
            IFK = (FK(J) - FK(J-1)) * (ROT - RT(J-1))
     &            / (RT(J) - RT(J-1)) + FK(J-1)
         ENDIF
   50 CONTINUE

      KOFF = LAMW * (1.0 + 3.0 * IFK * WOFF)

      IF (IOUT .EQ. 25) THEN
         WRITE(8,*) 'DI = ', DI, ' RI = ', RI
         WRITE(8,*) 'THIC = ', THIC, ' ROT = ', ROT
         WRITE(8,*) 'IFK = ', IFK, ' WOFF = ', WOFF
         WRITE(8,*) 'LAMW = ', LAMW, ' KOFF = ', KOFF
      ENDIF

C     CALCULATE THE CROSS-SECTIONAL AREA AND MOMENT OF INERTIA

      AREA = PI * ((RI + THIC) ** 2 - RI ** 2)
      MI = PI * ((RI + THIC) ** 4 - RI ** 4) / 4.0

C     OBTAIN STRESS CONCENTRATION FACTORS AND RADII APPROPRIATE TO LOCATION
C     THIS IS THE INTERIOR SURFACE

      SKT1 = K(1,2)
      SKT2 = K(2,2)
      RO = RI + THIC

      IF (IOUT .EQ. 25) THEN
         WRITE(8,*) 'AREA = ', AREA, ' MI = ', MI
         WRITE(8,*) 'K(1,2) = ', K(1,2), ' SKT1 = ', SKT1
         WRITE(8,*) 'K(2,2) = ', K(2,2), ' SKT2 = ', SKT2
         WRITE(8,*) 'THIC = ', THIC, ' RO = ', RO
         WRITE(8,*) 'ALPHA = ', ALPHA, ' NU = ', NU
         WRITE(8,*) 'DLTAT = ', DLTAT, ' EM = ', EM
         WRITE(8,*)
      ENDIF

      RI2 = RI ** 2
      RO2 = RO ** 2
      RDIFF = RO2 - RI2

      GEOM = 1.00 - 2.00 * LOG (RO / RI) * RO2 / RDIFF

C     TEMPERATURE STRESS

      STHMA = ((EM * ALPHA * DLTAT) / (2.00 * (1.00 - NU)
```

```
     &        * LOG (RO / RI))) * GEOM
C     AXIAL STRESS CALCULATIONS

      STR1A = PSTAT / AREA
      STR1B = (MSTAT(1) * COS (ANGLE) + MSTAT(2) * SIN (ANGLE))
     &        * RI / MI
      STR1C = (PC - PCO) * RI2 / RDIFF

      STATIC(1) = (STR1A + STR1B + STR1C) * SKT1 * KOFF + STHMA

C     HOOP (2) AND RADIAL (3) STRESS CALCULATIONS

      STR2A = PC * (RI2 + RO2) / RDIFF
      STR2B = - 2.0 * PCO * RO2 / RDIFF

      STATIC(2) = (STR2A + STR2B) * SKT2 + STHMA

      STATIC(3) = - PC

C     SHEAR STRESS

      STATIC(4) = TSTAT * RI / (2.0 * MI) - (2.0 / AREA
     &            * (VSTAT(1) * COS (ANGLE) + VSTAT(2) * SIN (ANGLE)))


      IF (IOUT.EQ.25) THEN
         WRITE(8,*) 'RO2 = ', RO2, 'RI2 = ', RI2
         WRITE(8,*) 'RDIFF = ', RDIFF, ' GEOM = ', GEOM
         WRITE(8,*) 'STATIC STRESS VALUES'
         WRITE(8,*) 'AXIAL STRESSES'
         WRITE(8,*) '   STR1A = ', STR1A, ' STR1B = ', STR1B
         WRITE(8,*) '   STR1C = ', STR1C, ' STHMA = ', STHMA
         WRITE(8,*) '   STATIC(1) = ', STATIC(1)
         WRITE(8,*) 'HOOP STRESSES'
         WRITE(8,*) '   STR2A = ', STR2A, ' STR2B = ', STR2B,
     &              ' STHMA = ', STHMA
         WRITE(8,*) '   STATIC(2) = ', STATIC(2)
         WRITE(8,*) 'RADIAL STRESS', '   STATIC(3) = ', STATIC(3)
         WRITE(8,*) 'SHEAR STRESS', '   STATIC(4) = ', STATIC(4)
         WRITE(8,*)
      ENDIF

      DO 100 I = 1, NLOAD

C     AXIAL STRESS CALCULATIONS

         SIG1A(I) = P(I) / AREA
         SIG1B(I) = (M(1,I) * COS (ANGLE) + M(2,I) * SIN (ANGLE))
     &              * RI / MI

         STRAMP(1,I) = (SIG1A(I) + SIG1B(I)) * SKT1 * KOFF

C        HOOP (2) AND RADIAL (3) STRESSES ARE ZERO
C        BECAUSE PRESSURES ARE CONSTANT

         STRAMP(2,I) = 0.0

         STRAMP(3,I) = 0.0

C        SHEAR STRESS

         STRAMP(4,I) = T(I) * RI / (2.0 * MI) - (2.0 / AREA *
     &                 (V(1,I) * COS (ANGLE) + V(2,I) * SIN (ANGLE)))


         IF (IOUT.EQ.25) THEN
            WRITE(8,*) 'STRESS VALUES FOR I = ',I
            WRITE(8,*) 'AXIAL STRESSES'
            WRITE(8,*) '   SIG1A = ', SIG1A(I), ' SIG1B = ', SIG1B(I)
            WRITE(8,*) '   STRAMP(1,I) = ', STRAMP(1,I)
            WRITE(8,*) 'HOOP STRESSES', '   STRAMP(2,I) = ',STRAMP(2,I)
            WRITE(8,*) 'RADIAL STRESS', '   STRAMP(3,I) = ',STRAMP(3,I)
            WRITE(8,*) 'SHEAR STRESS', '   STRAMP(4,I) = ', STRAMP(4,I)
            WRITE(8,*)
```

```
          ENDIF
  100 CONTINUE

        IF (IOUT .EQ. 25) THEN
           WRITE(8,*) 'I      AXIAL      HOOP      RADIAL      SHEAR'
           WRITE(8,*) STATIC(1), STATIC(2), STATIC(3), STATIC(4)
           DO 300 I = 1, NLOAD
              WRITE(8,*) I, STRAMP(1,I), STRAMP(2,I), STRAMP(3,I),
     &                      STRAMP(4,I)
  300      CONTINUE
        ENDIF

        RETURN
        END

C***********************************************************************
C
C   SUBROUTINE CYCOUN CALCULATES THE NUMBER OF CYCLES BY RAINFLOW
C   COUNTING, CREATES A STRESS VS. CYCLES TABLE, AND DETERMINES
C   THE EQUIVALENT MEAN STRESS
C
C   PROGRAMMER:  S. SUTHARSHANA, L. NEWLIN
C         DATE:  DECEMBER 1992
C      VERSION:  92.5
C
C***********************************************************************

        SUBROUTINE CYCOUN (DSALT, E, EM, NBIN, NEUB, NUMSEG, M,
     &                     SE, SPR, SM, TRUNC)

C   INPUTS: E, EM, M, NEUB, NUMSEG, SE, SPR, TRUNC
C
C
C   OUTPUTS: SPR, DSALT, NBIN, SM

C   SUBPROGRAM: NEUBER

C       IMPLICIT NONE

        COMMON  IOUT

        INTEGER  MAXM, MAXSEG

        PARAMETER (MAXM = 20000, MAXSEG = 10)

        INTEGER  I, INDEX(MAXM), IOUT, J, JMAX, K, M, N,
     &           NBIN(100), NEWTOT, NUMSEG, OVER

        INTEGER  INEUB, IRET, KGROW, KPROB

        REAL     DSALT, E(MAXSEG), EE(MAXM), EM, HIGH,
     &           LOW, NEUB, NEUBER, S(MAXM), SALTF,
     &           SE(MAXSEG), SPR(MAXM), SEFMAX,
     &           SM, SMEANF, SMAX, SMIN, SP(MAXM),
     &           TEST1(MAXM), TEST2(MAXM), TRUNC

        COMMON/CNTRL/INEUB, IRET, KGROW, KPROB

C                         LIST OF VARIABLES
C
C input variables:
C
C   E()           1-D ARRAY CONTAINING THE STRAIN VALUES
C   EM()          YOUNG'S MODULUS BEFORE YIELD
C   IOUT          OUTPUT DUMP CONTROLLER
C   M             TOTAL NUMBER OF STRESS DATA POINTS PER PERIOD
C   MAXM          MAXIMUM NUMBER OF POINTS IN STRESS TIME HISTORY
C   MAXSEG        MAXIMUM NUMBER OF SEGMENTS ALLOWED
C   NUMREG        NUMBER OF REGIONS OF INTEREST
C   NUMSEG        NUMBER OF SEGMENTS OF INTEREST
C   SE()          1-D ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS
```

```
C  SPR(M)        PRINCIPAL STRESS HISTORY
C  TRUNC         VALUE USED TO FILTER OUT NOISE
C
C intermediate variables:
C
C  EE()          HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
C  INDEX()       COUNTER FOR EFFECTIVE STRESSES
C  I,J,K         COUNTERS FOR VARIOUS DO LOOPS
C  JMAX          INDEX (LOCATION) OF SEFMAX IN SPR()
C  N             NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
C  NEUB          NEUBER'S RULE MODEL ACCURACY FACTOR
C  NEUBER        FUNCTION TO CALCULATE EQUIVALENT MEAN STRESS
C  NEWTOT        TOTAL NUMBER OF EFFECTIVE STRESS VALUES AFTER FILTERING
C  OVER          FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
C  S(NEWTOT)     FILTERED EFFECTIVE STRESSES
C  SALTF         ALTERNATING STRESS FOR LARGEST STRESS RANGE CYCLE
C  SEFMAX        LARGEST EFFECTIVE STRESS
C  SM            SM = EQUIVALENT MEAN STRESS
C  SMEANF        MEAN STRESS FOR LARGEST STRESS RANGE CYCLE
C  SP(M+1)       RESEQUENCED EFFECTIVE STRESSES; # OF PTS = M+1
C  TEST1()       1-D ARRAY USED IN FILTERING THE STRESSES
C  TEST2()       1-D ARRAY USED IN FILTERING THE STRESSES

C  dump input data
      if (iout.eq.30) then
         write(8,*)' cycoun inputs'
         write(8,*)'m     :',m
         WRITE(8,*) 'EM  :',EM,'     TRUNC    :',TRUNC,'    NEUB  :',NEUB
         WRITE(8,*) 'E():', E
         WRITE(8,*) 'SE():', SE
         WRITE(8,*) 'NUMSEG:', NUMSEG
         write(8,*)' '
      endif

C  INITIALIZE ARRAYS

      DO 50 I = 1, MAXM
         SP(I) = 0.0
         S(I) = 0.0
         EE(I) = 0.0
         INDEX(I) = 0
         TEST1(I) = 0.0
         TEST2(I) = 0.0
   50 CONTINUE

      SM = 0.0

C********************* B E G I N    R E S E Q U E N C E ********************
C  RESEQUENCE effective stresses (needed for rainflow analysis);
C   largest effective stress is placed at beginning and end of SP(M+1)

C find SEFMAX, the largest sigma,eff, and JMAX, its location within SPR(M)

      SEFMAX = -1.0E+20
      DO 200 I=1,M
         IF ( SPR(I) .GT. SEFMAX ) THEN
            SEFMAX = SPR(I)
            JMAX = I
         ENDIF
  200 CONTINUE

C   assign all points from JMAX out, to the beginning of SP()
      DO 210 I = 1, M-JMAX+1
         J = JMAX-1 + I
         SP(I) = SPR(J)
  210 CONTINUE

C   assign points before JMAX to the end of SP()
      J = 0
      DO 220 I = M-JMAX+2, M
         J = J + 1
         SP(I) = SPR(J)
  220 CONTINUE
      SP(M+1) = SPR(JMAX)
      if (iout.eq.30) then
```

```
          write(8,*)'sefmax:',sefmax,'     jmax:',jmax
          write(8,*)'sp(m+1):',(sp(i),i=1,m+1)
      endif

C********************** E N D   R E S Q U E N C E *************************

C*********************** B E G I N   F I L T E R ***************************
C  Filter the resequenced effective stresses, leaving only peaks and
C  valleys (excursions larger than TRUNC are deleted during rainflow
C  counting) in (NEWTOT), where NEWTOT is the new number of points
C

      DO 300 I = 2, M
         TEST1(I) = SP(I-1) - SP(I)
         TEST2(I) = TEST1(I) * (SP(I) - SP(I+1))
  300 CONTINUE

c       if (iout .eq. 30) then
c          do 305 i = 2, m
c             write(8,*) 'test1 = ', test1(i), ' test2 = ', test2(i)
c  305     continue
c       endif

      K = 1
      INDEX(1) = 1

      DO 310 I = 2, M
         IF ((TEST1(I) .NE. 0) .AND. (TEST2(I) .LT. 0)) THEN
            K = K + 1
            INDEX(K) = I
         ENDIF
  310 CONTINUE

      NEWTOT = K + 1
      INDEX(NEWTOT) = M + 1

      DO 320 I = 1, NEWTOT
         K = INDEX(I)
         S(I) = SP(K)
  320 CONTINUE

      if (iout.eq.30) then
         write(8,*)'newtot:',newtot
         write(8,*)'s(newtot):',(s(i),i=1,newtot)
      endif

C************************** E N D   F I L T E R ***************************

C********************** B E G I N   R A I N F L O W ***********************
C  RAINFLOW ANALYSIS to identify cycles within effective stress data,
C  S(NEWTOT); places each cycle's max and min values into SPR(N)
C
C  counters: I counts # of cycles found, J counts how many S()'s counted,
C            K accumulates unmatched points

      I = 0
      J = 0
      K = 0
  400 CONTINUE

      J = J+1
      K = K+1

C   check J to avoid reading beyond end of filtered stress data
      IF ( J .GT. NEWTOT ) GOTO 499

C   read stress point into a holding array to be checked for cycles
      EE(K) = S(J)

  410 IF ( K .LT. 3 ) GOTO 400
      IF (ABS (EE(K) - EE(K-1)) .LT. ABS (EE(K-1) - EE(K-2))) GOTO 400

C    if not, then a cycle has been found, but we need to check for truncation
```

```
      IF (ABS (EE(K-1) - EE(K-2)) .GT. TRUNC) THEN
C        cycle is large enough to save
         I = I+1
         SMAX = AMAX1( EE(K-1), EE(K-2) )
         SMIN = AMIN1( EE(K-1), EE(K-2) )
         SMEANF = (SMAX + SMIN)/2.0
         SPR(I) = SMAX - SMEANF
      ENDIF
C   discard points K-1 and K-2, and decrement the counter of unmatched points
      EE(K-2) = EE(K)
      K = K-2
C   return for more counting
      GOTO 410
  499 CONTINUE
C   N equals the final number of cycles found
      N = I
      if (iout.eq.25) then
         write(8,*)'N :',n
         write(8,*)'spr(n):'
         do 12 i=1,n
            write(8,*) spr(i)
   12    continue
      endif
      IF (N .EQ. 0) THEN
C TRUNCATION FILTER TOO LARGE -- NO CYCLES LEFT
         GOTO 710
      ENDIF
C
C********************** E N D   R A I N F L O W **************************
C  calculate alternating and mean effective stresses for the largest
C  stress cycle
      SALTF = SPR(N)
C  Assign the stress cycles to bins
      DSALT = SALTF/100.0
      LOW = SALTF + DSALT/2.0
      DO 510 I= 1, 100
         HIGH = LOW
         LOW = LOW - DSALT
         NBIN(I) = 0
         DO 500 J=1, N
            IF( (SPR(J) .GT. LOW) .AND. (SPR(J) .LT. HIGH) ) THEN
               NBIN(I) = NBIN(I) + 1
            ENDIF
  500    CONTINUE
  510 CONTINUE
      if (iout.eq.25) then
         write(8,*)'saltf :',saltf
         write(8,*)'smeanf:',smeanf
      endif
C******* Determine Equivalent Mean Stress, SM(N) for the largest cycle ****
C
      OVER = 0
C  We are calculating the equivalent mean stress using neuber's rule
C  SM is the equivalent mean stress
```

```
      IF (INEUB .EQ. 1) THEN
         SM = NEUBER (EM, SALTF, SMEANF, NUMSEG, E, SE, NEUB, OVER)
         if (iout.eq.25) write(8,*)'sm : ', sm
      ELSE
         SM = SMEANF
      ENDIF

  710 CONTINUE

      RETURN
      END
C***************************************************************************
C  NEUBER USES NEUBERS RULE AND THE STRESS-STRAIN CURVE TO CALCULATE THE
C  THE MEAN STRESS.  PROGRAM ASSUMES THAT THE STRESS STRAIN CURVE IS
C  PIECEWISE LINEAR WITH AT MOST FIVE SECTIONS.
C
C  PROGRAMER: L. NEWLIN
C       DATE: 13SEP88
C    VERSION: 92.1, 92.2, 92.3, 92.4, 92.5
C***************************************************************************

      FUNCTION NEUBER (EM, SALT, SMEAN, NUMSEG, E, SE, NEUB, OVER)

C  INPUTS:   EM, SALT, SMEAN, NUMSEG, SE, E, NEUB, OVER
C  OUTPUTS:  NEUBER

C      IMPLICIT NONE

      COMMON IOUT

      INTEGER  I, IOUT, MAXSEG, NUMSEG, OVER

      PARAMETER (MAXSEG = 10)

      REAL      E(MAXSEG), EM, EPSLON, NEUB, NEUBER, PRODCT, SALT,
     &          SE(MAXSEG), SMEAN, ST, TEMP
C
C                          LIST OF VARIABLES
C
C  E()          STRAIN VALUES FOR EACH SEGMENT
C  EM           YOUNG'S MODULUS BEFORE YIELD
C  EPSLON       CALCULATED STRAIN (WHERE PLASTIC=ELASTIC DEFORMATION)
C  I            CONTROLS DO LOOP FOR EACH SEGMENT
C  IOUT         OUTPUT DUMP CONTROLLER
C  MAXSEG       MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN)
C  NEUB         NEUBER'S RULE MODEL ACCURACY FACTOR
C  NEUBER       TOTAL EQUIVALENT MEAN STRESS
C  NUMSEG       NUMBER OR SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
C  OVER         FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
C  PRODCT       STRESS STRAIN PRODUCT (WHERE PLASTIC=ELASTIC DEFORMAITON)
C  SALT         TOTAL ALTERNATING STRESS
C  SE()         1-DIMENSIONAL ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS
C                 FOR A MULTI-SEGMENT CURVE
C  SMEAN        MEAN STRESS
C  ST           UNI-AXIAL TOTAL STRESS
C  TEMP         TEMPORARY VARIABLE FOR NEUBER

      TEMP = 0.00
      ST =  SALT * SMEAN / (ABS (SMEAN)) + SMEAN
      PRODCT = NEUB * (ST ** 2) / EM

      IF (PRODCT .LE. SE(1)) THEN

         TEMP = SMEAN

      ELSE

         DO 800 I = 1, (NUMSEG - 1)
            IF (PRODCT .GT. SE(I)) THEN
               IF (PRODCT .LT. SE(I+1)) THEN
                  EPSLON = E(I) + ((E(I+1) - E(I)) /
     &                     (SE(I+1) - SE(I))) * (PRODCT - SE(I))
                  TEMP = PRODCT / EPSLON - SALT
```

```
            ENDIF
         ENDIF
 800    CONTINUE

      ENDIF

      IF (ABS(TEMP) .LT. 1.0E-04) THEN
         OVER = 1
         WRITE(8,*) 'THE VALUE PRODCT EXCEEDED STRESS-STRAIN CURVE'
      ENDIF

      TEMP = TEMP * ABS(ST) / ST
      NEUBER = TEMP

      IF (IOUT .EQ. 25) THEN
         WRITE(8,*)
         WRITE(8,*) 'VALUES FROM NEUBER'
         WRITE(8,*) 'INPUT VALUES'
         WRITE(8,*) '   EM = ', EM, ' NEUB = ', NEUB, ' OVER = ', OVER
         WRITE(8,*) '   SALT = ', SALT, ' SMEAN = ', SMEAN
         WRITE(8,*) 'CALCULATED VALUES'
         WRITE(8,*) '   ST = ', ST, ' PRODUCT = ', PRODCT
         WRITE(8,*) '   EPSLON = ', EPSLON, ' NEUBER = ', TEMP
      ENDIF

      RETURN
      END
C***********************************************************************
C SUBROUTINE HYPDRW PERFORMS THE RANDOM VARIABLE DRAWS IN THE OUTER
C HYPERPARAMETER LOOP
C
C PROGRAMMER : S. SUTHARSHANA
C DATE : 25 JAN 1989
C VERSION : 92.1, 92.2, 92.3, 92.4, 92.5
C
C  Copyright (C) 1991, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      SUBROUTINE HYPDRW (AERD, AERS, ASTR, DSTR, KLAM, LAMGR, LAMKC,
     &                   LAMKH, LAMW, NEUB, SSTR, MVAR)

C  SUBPROGRAMS: PRYRV, RANDOM

C      IMPLICIT NONE

      INTEGER INEUB, IRET, KGROW, KPROB

      INTEGER IOUT

      DOUBLE PRECISION RAND

      REAL      AERD, AERDA, AERDB, AERS, AERSA, AERSB, AIA, AIB,
     &          AIR1, AIR2, AIT1, AIT2, AIR, AIT,
     &          AOCA, AOCB, AOCR1, AOCR2, AOCT1, AOCT2, AOCR, AOCT,
     &          ASTR, ASTRA, ASTRB,
     &          DPCMU, DPCSIG, DSTR, DSTRA, DSTRB, DTIMU, DTISIG,
     &          DTOMU, DTOSIG, DUM, INDIAA, INDIAB,
     &          INDIR, INDIR1, INDIR2, INDIT, INDIT1, INDIT2,
     &          KLAM, KLAMA, KLAMB, LAMKH, LAMKHA, LAMKHB,
     &          LAMKC, LAMKCA, LAMKCB, LAMGR, LAMGRA, LAMGRB,
     &          LAMNA, LAMNB, LAMNC, LAMND,
     &          LAMNK, LAMNMU, LAMNSG, LAMSA, LAMSB, LAMSC, LAMSD,
     &          LAMSK, LAMSMU, LAMSSG,
     &          LAMW, LAMWA, LAMWB, MVAR, MVARA, MVARB

      REAL      NEUB, NEUBA, NEUBB, NORM, PCMU, PCMUA, PCMUB, PCSIG,
     &          PCSIGA, PCSIGB, SSTR, SSTRA,
     &          SSTRB, TEST, THICA, THICB,
     &          THICR, THICR1, THICR2, THICT, THICT1, THICT2, TIMU,
     &          TIMUA, TIMUB, TISIG, TISIGA, TISIGB, TOMU, TOMUA, TOMUB,
     &          TOSIG, TOSIGA, TOSIGB, WITHA,
     &          WITHB, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
```

```
&          WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO, WOFFR,
&          WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFT, WOFFT1, WOFFT2,
&          WOFFT3, WOFFT4

      COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AIA,
     &    AIB, AIR1, AIR2, AIT1, AIT2, AIR, AIT,
     &    AOCA, AOCB, AOCR1, AOCR2, AOCT1, AOCT2, AOCR, AOCT,
     &    ASTRA, ASTRB,
     &    DPCMU, DPCSIG, DSTRA, DSTRB, DTIMU, DTISIG, DTOMU, DTOSIG,
     &    INDIAA,INDIAB,INDIR1, INDIR2, INDIR, INDIT1, INDIT2, INDIT,
     &    KLAMA, KLAMB, LAMGRA, LAMGRB, LAMKHA, LAMKHB, LAMKCA, LAMKCB,
     &    LAMNA, LAMNB, LAMNC,  LAMND, LAMNMU, LAMNSG,
     &    LAMSA, LAMSB, LAMSC,  LAMSD, LAMSMU, LAMSSG,
     &    LAMWA, LAMWB, MVARA, MVARB, NEUBA,  NEUBB,
     &    PCMU, PCMUA, PCMUB, PCSIG, PCSIGA, PCSIGB,
     &    RAND,
     &    SSTRA, SSTRB,
     &    THICA, THICB, THICR1, THICR2, THICR, THICT1, THICT2, THICT,
     &    TIMU, TIMUA, TIMUB, TISIG, TISIGA, TISIGB,
     &    TOMU, TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
     &    WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
     &    WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
     &    WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFR, WOFFT1, WOFFT2,
     &    WOFFT3, WOFFT4, WOFFT

      COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
      COMMON IOUT

C  START MAKING THE RANDOM DRAWS

      CALL PRYRV (RAND, AIR1, AIR2, AIT1, AIT2, AIR, AIT)

      IF (KPROB .EQ. 1) THEN
         CALL RANDOM (TEST, RAND)
         IF (TEST .LE. WOFFE) THEN
            CALL PRYRV (RAND, WOFFR1, WOFFR2, WOFFT1, WOFFT2, WOFFR,
     &                  WOFFT)
            WOFFLO = WOFFA
            WOFFHI = WOFFB
         ELSE
            CALL PRYRV (RAND, WOFFR3, WOFFR4, WOFFT3, WOFFT4, WOFFR,
     &                  WOFFT)
            WOFFLO = WOFFC
            WOFFHI = WOFFD
         ENDIF
         IF (IOUT .EQ. 15) THEN
            WRITE(8,*) 'TEST =', TEST, ' WOFFE =', WOFFE
            WRITE(8,*) 'WOFFLO =', WOFFLO, ' WOFFHI =', WOFFHI
         ENDIF

         CALL PRYRV (RAND, INDIR1, INDIR2, INDIT1, INDIT2, INDIR, INDIT)
         CALL PRYRV (RAND, THICR1, THICR2, THICT1, THICT2, THICR, THICT)
         CALL PRYRV (RAND, AOCR1, AOCR2, AOCT1, AOCT2, AOCR, AOCT)
         CALL PRYRV (RAND, AERDA, AERDB, AERSA, AERSB, AERD, AERS)
         CALL PRYRV (RAND, ASTRA, ASTRB, DUM, DUM, ASTR, DUM)
      ELSEIF (KPROB .EQ. 2) THEN
         CALL PRYRV (RAND, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT)
      ENDIF

      CALL PRYRV (RAND, LAMNA, LAMNB,  LAMSA,  LAMSB,  LAMNK, LAMSK)
      LAMNMU = LAMND / (1.0 + LAMNK * LAMNC)
      LAMNSG = LAMNC / (1.0 + LAMNK * LAMNC)
      LAMSMU = LAMSD / (1.0 + LAMSK * LAMSC)
      LAMSSG = LAMSC / (1.0 + LAMSK * LAMSC)
      IF (IOUT .EQ. 15) THEN
         WRITE(8,*) 'LAMNK = ', LAMNK, ' LAMNMU = ', LAMNMU,
     &              ' LAMNSG = ', LAMNSG
         WRITE(8,*) 'LAMSK = ', LAMSK, ' LAMSMU = ', LAMSMU,
     &              ' LAMSSG = ', LAMSSG
      ENDIF

      IF (KPROB .EQ. 1) THEN
         CALL RANDOM (NORM, RAND)
         TIMU = TIMUA + NORM * DTIMU
```

```
            TISIG = TISIGA + NORM * DTISIG
            TOMU = TOMUA + NORM * DTOMU
            TOSIG = TOSIGA + NORM * DTOSIG
            PCMU = PCMUA + NORM * DPCMU
            PCSIG = PCSIGA + NORM * DPCSIG
         ENDIF

         IF (IOUT .EQ. 15) THEN
            WRITE(8,*) 'NORM = ', NORM
            WRITE(8,*) '   TIMU = ', TIMU, ' TISIG = ', TISIG
            WRITE(8,*) '   TOMU = ', TOMU, ' TOSIG = ', TOSIG
            WRITE(8,*) '   PCMU = ', PCMU, ' PCSIG = ', PCSIG
         ENDIF

         CALL PRYRV (RAND, LAMKHA, LAMKHB, LAMKCA, LAMKCB, LAMKH, LAMKC)
         CALL PRYRV (RAND, LAMWA, LAMWB, NEUBA, NEUBB, LAMW, NEUB)
         CALL PRYRV (RAND, DSTRA, DSTRB, KLAMA, KLAMB, DSTR, KLAM)
         CALL PRYRV (RAND, SSTRA, SSTRB, DUM, DUM, SSTR, DUM)
         CALL PRYRV (RAND, LAMGRA, LAMGRB, MVARA, MVARB, LAMGR, MVAR)
         LAMGR = EXP(LAMGR)
         RETURN
         END

C*************************************************************************
C SUBROUTINE PARDRW PERFORMS THE RANDOM LIFE DRIVER PARAMETER DRAWS IN
C THE INNER LOOP
C
C PROGRAMMER : S. SUTHARSHANA
C DATE : DECEMBER 1992
C VERSION : 92.5
C
C  Copyright (C) 1991, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***************************************************************************

         SUBROUTINE PARDRW (AI, CI, DLTAT, INDIA, LAMN, LAMS, PC,
        &                   THIC, WIDTH, WOFF)

C  SUBPROGRAMS: BETAGN, NORMGN

C        IMPLICIT NONE

         INTEGER INEUB, IRET, KGROW, KPROB

         INTEGER IOUT

         DOUBLE PRECISION RAND

         REAL      AERDA, AERDB, AERSA, AERSB, AI, AIA, AIB, AIR1, AIR2,
        &          AIT1, AIT2, AIR, AIT, AOC, AOCA, AOCB, AOCR1, AOCR2,
        &          AOCT1, AOCT2, AOCR, AOCT, ASTRA, ASTRB,
        &          CI, DLTAT, DPCMU, DPCSIG, DSTRA, DSTRB, DTIMU, DTISIG,
        &          DTOMU, DTOSIG, INDIA, INDIAA,
        &          INDIAB, INDIR, INDIR1, INDIR2, INDIT, INDIT1, INDIT2,
        &          KLAMA, KLAMB, LAMKHA, LAMKHB,
        &          LAMKCA, LAMKCB, LAMGRA, LAMGRB,
        &          LAMN, LAMNA, LAMNB, LAMNC, LAMND,
        &          LAMNMU, LAMNSG, LAMS, LAMSA, LAMSB, LAMSC, LAMSD,
        &          LAMSMU, LAMSSG, LAMWA, LAMWB, MVARA, MVARB

         REAL      NEUBA, NEUBB, PC, PCMU, PCMUA, PCMUB, PCSIG,
        &          PCSIGA, PCSIGB, SSTRA, SSTRB,
        &          THIC, THICA, THICB, THICR, THICR1, THICR2, THICT, THICT1,
        &          THICT2, TIMU, TIMUA, TIMUB, TIN, TISIG, TISIGA, TISIGB,
        &          TOMU, TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB, TOUT,
        &          WIDTH, WITHA, WITHB, WITHR1, WITHR2,
        &          WITHT1, WITHT2, WITHR, WITHT, WOFF,
        &          WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
        &          WOFFR, WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFT, WOFFT1,
        &          WOFFT2, WOFFT3, WOFFT4

         COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AIA,
        &     AIB, AIR1, AIR2, AIT1, AIT2, AIR, AIT,
        &     AOCA, AOCB, AOCR1, AOCR2, AOCT1, AOCT2, AOCR, AOCT,
```

```
     &     ASTRA, ASTRB,
     &     DPCMU, DPCSIG, DSTRA, DSTRB, DTIMU, DTISIG, DTOMU, DTOSIG,
     &     INDIAA,INDIAB,INDIR1, INDIR2, INDIR, INDIT1, INDIT2, INDIT,
     &     KLAMA, KLAMB, LAMGRA, LAMGRB, LAMKHA, LAMKHB, LAMKCA, LAMKCB,
     &     LAMNA, LAMNB, LAMNC,  LAMND, LAMNMU, LAMNSG,
     &     LAMSA, LAMSB, LAMSC,  LAMSD, LAMSMU, LAMSSG,
     &     LAMWA, LAMWB, MVARA, MVARB, NEUBA,  NEUBB,
     &     PCMU, PCMUA, PCMUB, PCSIG, PCSIGA, PCSIGB,
     &     RAND,
     &     SSTRA, SSTRB,
     &     THICA, THICB, THICR1, THICR2, THICR, THICT1, THICT2, THICT,
     &     TIMU, TIMUA, TIMUB, TISIG, TISIGA, TISIGB,
     &     TOMU, TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
     &     WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
     &     WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
     &     WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFR, WOFFT1, WOFFT2,
     &     WOFFT3, WOFFT4, WOFFT

      COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
      COMMON IOUT

      CALL BETAGN (RAND, AIR, AIT, AIA,  AIB,  AI)
      IF(KPROB .EQ. 1) THEN
         CALL BETAGN (RAND, WOFFR, WOFFT, WOFFLO, WOFFHI, WOFF)
         CALL BETAGN (RAND, INDIR, INDIT, INDIAA, INDIAB, INDIA)
         CALL BETAGN (RAND, THICR, THICT, THICA,  THICB,  THIC)
         CALL BETAGN (RAND, AOCR, AOCT, AOCA, AOCB, AOC)
         CI = AI/AOC
      ELSEIF(KPROB .EQ. 2) THEN
         CALL BETAGN (RAND, WITHR, WITHT, WITHA, WITHB, WIDTH)
      ENDIF
      CALL NORMGN (RAND, LAMNMU, LAMNSG, LAMN)
      CALL NORMGN (RAND, LAMSMU, LAMSSG, LAMS)

      IF (KPROB .EQ. 1) THEN
         CALL NORMGN (RAND, TIMU, TISIG, TIN)
         CALL NORMGN (RAND, TOMU, TOSIG, TOUT)
         DLTAT = TIN - TOUT
         CALL NORMGN (RAND, PCMU, PCSIG, PC)
      ENDIF

      IF (IOUT .EQ. 15) THEN
         WRITE(8,*) 'AI =',AI,' AOC =',AOC,' CI =',CI
         WRITE(8,*) 'LAMN =',LAMN,' LAMS =',LAMS
         WRITE(8,*) 'THIC =', THIC
         WRITE(8,*) 'INDIA =',INDIA,' PC =',PC
         WRITE(8,*) 'TIN =',TIN,' TOUT =',TOUT,' DLTAT =',DLTAT
         WRITE(8,*) 'WOFF =',WOFF,' WIDTH =',WIDTH
      ENDIF

      RETURN
      END
C*************************************************************************
C SUBROUTINE STRIF1 CALCULATES THE STRESS INTENSITY FACTOR
C IN A FINITE WIDTH PLATE WITH AN ELLIPTIC FLAW
C
C PROGRAMMER : S. SUTHARSHANA
C
C SIF EXPRESSIONS ADAPTED FROM NAS8609A.FOR VERSION OF NASA/FLAGRO CODE
C EXPRESSIONS SURFACE CRACK IN A FINITE WIDTH PLATE IS EMPLOYED
C ..  THE WIDTH IS SET TO THE CIRCUMFERENCE OF THE TUBE
C
C DATE : DECEMBER 1992
C VERSION 92.5
C*************************************************************************

      SUBROUTINE STRIF1(A, C, F0, F2, INDIA, THIC)

C
C        A      - CRACK DEPTH, A
C        C      - CRACK LENGTH, C
C
C  INTERNAL VARIABLES
C
C        AOC75  - EVENTUALLY, (A/T)**.75
```

```
C         AOT     - A/T
C         AOTSQ   - (A/T)**2
C         C       - CRACK LENGTH, C
C         COA     - C/A
C         COA4    - (C/A)**4
C         CONST   - SQRT(PI*A)
C         PI      - 3.14159...
C         POWR    - POWR TO BE RAISED
C         SQCOA   - SQRT(C/A)
C         TRM     - 1 - A/C
C
C      IMPLICIT NONE

      INTEGER IOUT

      REAL A, AOC, AOC75, AOT, AOTSQ, C, COA, COA4, EM0,
     &     EM1, EM2, EM3, F0(2), F2(2), FPA, FPC, FW, G1, G2,
     &     GA, GC, H1, H2, INDIA, PI, Q,
     &     SQCOA, THIC, WIDTH

      REAL*8 POWR, TRM

      COMMON IOUT

      DATA PI/3.14159265358979/

      WIDTH = PI * (INDIA + THIC)
      AOC =A/C
      AOT =A/THIC
      IF(IOUT .EQ. 22) THEN
         WRITE(8,*) 'AOC, AOT', AOC, AOT
      ENDIF
C
C                                                 Compute SIFs
C
      IF (AOC.LE.1.0) THEN
         G1    = -1.22-0.12*AOC
         AOC75 = SQRT(AOC)
         AOC75 = AOC75 * SQRT(AOC75)
         G2    = 0.55 + AOC75 * ((-1.05) + AOC75*0.47)
         H1    = 1.0 - AOT * (0.34 + 0.11*AOC)
         H2    = 1.0 + AOT*(G1 + G2*AOT)
         EM1   = 1.13 - 0.09*AOC
         EM2   = -0.54 + 0.89 / (0.2 + AOC)
         TRM   = 1.0-AOC
         IF (ABS(TRM).GT.0.001D0) THEN
            POWR = TRM**24
         ELSE
            POWR = 0D0
         ENDIF
         EM3   = 0.5 - 1.0/(0.65+AOC) + 14.0*POWR
         AOTSQ = AOT*AOT
         EM0   = EM1 + AOTSQ*(EM2 + AOTSQ*EM3)
         Q     = 1.0 + 1.464*AOC**1.65
         FPA    = 1.0
         FPC    = SQRT(AOC)
         GA     = 1.0
         GC     = 1.0 + (0.1 + 0.35*AOT*AOT)
C
      ELSE
C
C                                                        AOC > 1
C
         COA   = 1.0/AOC
         SQCOA = SQRT(COA)
         EM1   = (1.0+0.04*COA)*SQCOA
         COA4  = COA*COA*COA*COA
         EM2   = 0.2*COA4
         EM3   = -0.11*COA4
         AOTSQ = AOT*AOT
         EM0   = EM1 + AOTSQ*(EM2 + AOTSQ*EM3)
         GA    = 1.0
         GC     = 1.0 + (0.1 +  0.35*AOTSQ*COA)
         AOC75 = SQCOA*SQRT(SQCOA)
         H1    = 1.0 + AOT*(-0.4 - 0.41*COA + AOT*(0.55 +
```

```
&                 AOC75*(-1.93 + AOC75*1.38)))
      H2      = 1.0 + AOT*(-2.11 + 0.77*COA + AOT*
&                 (0.55 + AOC75*(-0.72 + AOC75*0.14) ) )
      Q       = 1.0 + 1.464*COA**1.65
      FPA     = SQRT(COA)
      FPC     = 1.0
    ENDIF
C
C   TREAT IT AS AN EDGE CRACK IF 2C > WIDTH
C
      IF ( 2.0*C .LT. WIDTH ) THEN
        FW      = SQRT( 1.0/COS( SQRT(AOT)*PI*C/WIDTH ) )
      ELSE
        FW      = 1.0
      ENDIF
      IF(IOUT .EQ. 22) THEN
        WRITE(8,*) 'WIDTH, C, A, FW', WIDTH, C, A, FW
      ENDIF
C
C   for "a" direction
C
      F0(1)     = EM0*GA*FPA*FW/SQRT(Q)
      F2(1)     = F0(1)*H2
      IF(IOUT .EQ. 20) THEN
        WRITE(8,*)'A DIR F0, F2', F0(1), F2(1)
      ENDIF
C
C   for "c" direction
C
      F0(2)     = EM0*GC*FPC*FW/SQRT(Q)
      F2(2)     = F0(2)*H1
      IF(IOUT .EQ. 20) THEN
        WRITE(8,*)'C DIR F0, F2', F0(2), F2(2)
      ENDIF


      RETURN
      END
C*************************************************************************
C SUBROUTINE STRIF2 CALCULATES THE STRESS INTENSITY FACTOR
C IN A PLATE WITH A 2a CRACK
C
C PROGRAMMER : S. SUTHARSHANA
C
C SIF EXPRESSION FROM 'ELEMENTARY FRACTURE MECHANICS' BY D. BROEK
C
C DATE : 19 NOV 1989
C VERSION 92.1, 92.2, 92.3, 92.4, 92.5
C*************************************************************************

      SUBROUTINE STRIF2(A, F0, F2, WIDTH)
C     IMPLICIT NONE

      REAL A, F0(2), F2(2), PI, PIA, WIDTH

      DATA PI/3.14159265358979/

      PIA = PI * A

      F0(1) = SQRT( 1.0/COS(PIA/WIDTH) )
      F2(1) = F0(1)

      RETURN
      END
C*************************************************************************
C  SUBROUTINE GRODAT READS MATERIAL PROPERTIES AND PERFORMS REGRESSION ON
C  CRACK GROWTH DATA
C  PROGRAMMER:  S. SUTHARSHANA
C        DATE:  DECEMBER 1992
C     VERSION:  92.5
C
C  Copyright (C) 1991, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
```

```
C*****************************************************************************
      SUBROUTINE GRODAT (CEE, CO, DEE, DKTHO, EMM, ENN, FTY,
     &                   KC, PEE, QUE)

C      IMPLICIT NONE

      EXTERNAL DETER4

      INTEGER MAXDAT, MAXDIV

      PARAMETER (MAXDAT = 200, MAXDIV = 10)

      INTEGER IOUT, IREGOP, I, J, K, NDIV, NP(MAXDIV),
     &        NPR, NPRMN1

      REAL  AA, BB, CC, CEE, DADN(MAXDIV, MAXDAT),
     &      CO, DEE, DELK(MAXDIV, MAXDAT), DENOM, DETER4,
     &      DIFFX1, DIFFX2, DIFFX3, DIFFX4, DIFFY, DKTH, DKTHO,
     &      EMM, ENN, FTY, KC, LNCEE, MEANX1, MEANX2, MEANX3, MEANX4,
     &      MEANY, PEE, QUE, RDATA(MAXDIV), RKC,
     &      SX1Y, SX2Y, SX3Y, SX1X1, SX1X2, SX1X3, SX2X2, SX2X3,
     &      SX3X3, SX4Y, SX1X4, SX2X4, SX3X4, SX4X4, Y

C==================== DESCRIPTION OF LOCAL VARIABLES =================
C  DADN()   CRACK GROWTH RATE
C  DELK()   SIF RANGE
C  DENOM    DENOMINATOR
C  RDATA()  STRESS RATIOS
C  RKC      (1-R)Kc
C=====================================================================
      CHARACTER*40 DESCRP

      COMMON  IOUT

C READ THE da/dN vs. DK DATA FOR THE DIFFERENT REGIONS

      READ(1,*) DESCRP, FTY, KC, NDIV, IREGOP
      WRITE(3,6000) DESCRP, FTY, KC, NDIV, IREGOP

      IF(IREGOP.LT.0 .OR. IREGOP.GT.4) THEN
         WRITE(8,*) 'INVALID REGRESSION OPTION SPECIFICATION'
         CALL TRMNAT
      ENDIF

C READ THRESHOLD DESCRIPTION INFORMATION

      READ(1,*) DKTHO, CO, DEE
      WRITE(3, 6050) DKTHO, CO, DEE

      IF(IREGOP .EQ. 0) THEN
         READ(1,*) PEE
      ELSEIF(IREGOP .EQ. 1) THEN
         READ(1,*) EMM, PEE
      ELSEIF(IREGOP .EQ. 2) THEN
         READ(1,*) QUE, PEE
      ELSEIF(IREGOP .EQ. 3) THEN
         READ(1,*) EMM, QUE, PEE
      ENDIF

      DO 190 I = 1, NDIV
         READ(1,*)  NP(I), RDATA(I)
         WRITE(3,6150) RDATA(I)
         IF(NP(I) .GT. MAXDAT) THEN
            WRITE(8,*) 'NUMBER OF GROWTH RATE DATA POINTS PER DIVISION
     &                  EXCEEDED'
            CALL TRMNAT
         ENDIF
         DO 180 J = 1, NP(I)
            READ(1,*) DADN(I,J), DELK(I,J)
            WRITE(3,6200) DADN(I,J), DELK(I,J)
 180     CONTINUE
 190  CONTINUE

C=============================
```

```
C PERFORM REGRESSION ON THE DATA
C===============================

C  CALCULATE SX2, SY2, SXY

      IF(IREGOP .EQ. 0) THEN

            NPR = 0
            MEANY = 0.0
            MEANX1 = 0.0
            MEANX2 = 0.0
            MEANX3 = 0.0
            SX1Y = 0.0
            SX2Y = 0.0
            SX3Y = 0.0
            SX1X1 = 0.0
            SX1X2 = 0.0
            SX1X3 = 0.0
            SX2X2 = 0.0
            SX2X3 = 0.0
            SX3X3 = 0.0

            DO 275 J = 1, NDIV
               RKC = (1.-RDATA(J))*KC
               DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
               DO 250 K = 1, NP(J)
                  Y =  LOG10(DADN(J,K))
     &                 - PEE * LOG10(DELK(J,K) - DKTH)
                  MEANY = MEANY + Y
                  MEANX1 = MEANX1 + LOG10(DELK(J,K))
                  MEANX2 = MEANX2 + LOG10(1.-RDATA(J))
                  MEANX3 = MEANX3 - LOG10(RKC - DELK(J,K))
 250           CONTINUE
               NPR = NPR + NP(J)
 275        CONTINUE
            MEANY = MEANY/FLOAT(NPR)
            MEANX1 = MEANX1/FLOAT(NPR)
            MEANX2 = MEANX2/FLOAT(NPR)
            MEANX3 = MEANX3/FLOAT(NPR)
C
            NOW CALCULATE SY2, SX2, AND SXY

            DO 350 J= 1, NDIV
               RKC = (1.-RDATA(J))*KC
               DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
               DO 300 K = 1, NP(J)
                  Y = LOG10(DADN(J,K))
     &            - PEE * LOG10(DELK(J,K) - DKTH)
                  DIFFY = Y  - MEANY
                  DIFFX1 = LOG10(DELK(J,K)) - MEANX1
                  DIFFX2 = LOG10(1.-RDATA(J)) - MEANX2
                  DIFFX3 = - LOG10(RKC-DELK(J,K)) - MEANX3
                  SX1Y = SX1Y + DIFFX1 * DIFFY
                  SX2Y = SX2Y + DIFFX2 * DIFFY
                  SX3Y = SX3Y + DIFFX3 * DIFFY
                  SX1X1 = SX1X1 + DIFFX1 * DIFFX1
                  SX1X2 = SX1X2 + DIFFX1 * DIFFX2
                  SX1X3 = SX1X3 + DIFFX1 * DIFFX3
                  SX2X2 = SX2X2 + DIFFX2 * DIFFX2
                  SX2X3 = SX2X3 + DIFFX2 * DIFFX3
                  SX3X3 = SX3X3 + DIFFX3 * DIFFX3
 300           CONTINUE
 350        CONTINUE

            NPRMN1 = NPR - 1

            SX1Y = SX1Y/FLOAT(NPRMN1)
            SX2Y = SX2Y/FLOAT(NPRMN1)
            SX3Y = SX3Y/FLOAT(NPRMN1)
            SX1X1 = SX1X1/FLOAT(NPRMN1)
            SX1X2 = SX1X2/FLOAT(NPRMN1)
            SX1X3 = SX1X3/FLOAT(NPRMN1)
            SX2X2 = SX2X2/FLOAT(NPRMN1)
            SX2X3 = SX2X3/FLOAT(NPRMN1)
            SX3X3 = SX3X3/FLOAT(NPRMN1)
```

```
C CALCULATE THE COEFFICIENTS

          AA = SX2X2 * SX3X3 - SX2X3 ** 2
          BB = SX1X2 * SX3X3 - SX2X3 * SX1X3
          CC = SX1X2 * SX2X3 - SX2X2 * SX1X3

          DENOM = SX1X1 * AA - SX1X2 * BB + SX1X3 * CC

          ENN = ( SX1Y * AA - SX1X2 * (SX2Y * SX3X3 - SX2X3 * SX3Y)
     &          + SX1X3 * (SX2Y * SX2X3 - SX2X2 * SX3Y) ) / DENOM

          EMM = ( SX1X1 * (SX2Y * SX3X3 - SX2X3 * SX3Y) - SX1Y * BB
     &          + SX1X3 * (SX1X2 * SX3Y - SX2Y * SX1X3) ) / DENOM

          QUE = ( SX1X1 * (SX2X2 * SX3Y - SX2Y * SX2X3) + SX1Y * CC
     &          - SX1X2 * (SX1X2 * SX3Y - SX2Y * SX1X3) ) / DENOM

          LNCEE = MEANY-ENN*MEANX1-EMM*MEANX2-QUE*MEANX3
          CEE = 10.0**(LNCEE)

      ELSEIF(IREGOP .EQ. 1) THEN

          NPR = 0
          MEANY = 0.0
          MEANX1 = 0.0
          MEANX3 = 0.0
          SX1Y = 0.0
          SX3Y = 0.0
          SX1X1 = 0.0
          SX1X3 = 0.0
          SX3X3 = 0.0

          DO 1275 J = 1, NDIV
             RKC = (1.-RDATA(J))*KC
             DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
             DO 1250 K = 1, NP(J)
                Y =  LOG10(DADN(J,K))
     &               - EMM * LOG10(1.-RDATA(J))
     &               - PEE * LOG10(DELK(J,K) - DKTH)
                MEANY = MEANY + Y
                MEANX1 = MEANX1 + LOG10(DELK(J,K))
                MEANX3 = MEANX3 - LOG10(RKC - DELK(J,K))
 1250        CONTINUE
             NPR = NPR + NP(J)
 1275     CONTINUE
          MEANY = MEANY/FLOAT(NPR)
          MEANX1 = MEANX1/FLOAT(NPR)
          MEANX3 = MEANX3/FLOAT(NPR)

C         NOW CALCULATE SY2, SX2, AND SXY

          DO 1350 J= 1, NDIV
             RKC = (1.-RDATA(J))*KC
             DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
             DO 1300 K = 1, NP(J)
                Y =  LOG10(DADN(J,K))
     &               - EMM * LOG10(1.-RDATA(J))
     &               - PEE * LOG10(DELK(J,K) - DKTH)
                DIFFY = Y - MEANY
                DIFFX1 = LOG10(DELK(J,K)) - MEANX1
                DIFFX3 = - LOG10(RKC-DELK(J,K)) - MEANX3
                SX1Y = SX1Y + DIFFX1 * DIFFY
                SX3Y = SX3Y + DIFFX3 * DIFFY
                SX1X1 = SX1X1 + DIFFX1 * DIFFX1
                SX1X3 = SX1X3 + DIFFX1 * DIFFX3
                SX3X3 = SX3X3 + DIFFX3 * DIFFX3
 1300        CONTINUE
 1350     CONTINUE

          NPRMN1 = NPR - 1

          SX1Y = SX1Y/FLOAT(NPRMN1)
          SX3Y = SX3Y/FLOAT(NPRMN1)
          SX1X1 = SX1X1/FLOAT(NPRMN1)
```

```
            SX1X3 = SX1X3/FLOAT(NPRMN1)
            SX3X3 = SX3X3/FLOAT(NPRMN1)
C CALCULATE THE COEFFICIENTS
            DENOM = SX1X1 * SX3X3 - SX1X3 ** 2
            ENN = ( SX1Y * SX3X3 - SX1X3 * SX3Y ) / DENOM
            QUE = ( SX1X1 * SX3Y - SX1Y * SX1X3 ) / DENOM
            LNCEE = MEANY - ENN*MEANX1 - QUE*MEANX3
            CEE = 10.0**(LNCEE)
 1600     CONTINUE
       ELSEIF( IREGOP .EQ. 2 ) THEN
            NPR = 0
            MEANY = 0.0
            MEANX1 = 0.0
            MEANX2 = 0.0
            SX1Y = 0.0
            SX2Y = 0.0
            SX1X1 = 0.0
            SX1X2 = 0.0
            SX2X2 = 0.0
            DO 2275 J = 1, NDIV
               RKC = (1.-RDATA(J))*KC
               DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
               DO 2250 K = 1, NP(J)
                  Y = LOG10(DADN(J,K))
     &            - PEE * LOG10(DELK(J,K) - DKTH)
     &            + QUE * LOG10(RKC - DELK(J,K))
                  MEANY = MEANY + Y
                  MEANX1 = MEANX1 + LOG10(DELK(J,K))
                  MEANX2 = MEANX2 + LOG10(1.-RDATA(J))
 2250          CONTINUE
               NPR = NPR + NP(J)
 2275       CONTINUE
            MEANY = MEANY/FLOAT(NPR)
            MEANX1 = MEANX1/FLOAT(NPR)
            MEANX2 = MEANX2/FLOAT(NPR)
C           NOW CALCULATE SY2, SX2, AND SXY
            DO 2350 J= 1, NDIV
               RKC = (1.-RDATA(J))*KC
               DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
               DO 2300 K = 1, NP(J)
                  Y = LOG10(DADN(J,K))
     &            - PEE * LOG10(DELK(J,K) - DKTH)
     &            + QUE * LOG10(RKC - DELK(J,K))
                  DIFFY = Y  - MEANY
                  DIFFX1 = LOG10(DELK(J,K)) - MEANX1
                  DIFFX2 = LOG10(1.-RDATA(J)) - MEANX2
                  SX1Y = SX1Y + DIFFX1 * DIFFY
                  SX2Y = SX2Y + DIFFX2 * DIFFY
                  SX1X1 = SX1X1 + DIFFX1 * DIFFX1
                  SX1X2 = SX1X2 + DIFFX1 * DIFFX2
                  SX2X2 = SX2X2 + DIFFX2 * DIFFX2
 2300          CONTINUE
 2350       CONTINUE
            NPRMN1 = NPR - 1
            SX1Y = SX1Y/FLOAT(NPRMN1)
            SX2Y = SX2Y/FLOAT(NPRMN1)
            SX1X1 = SX1X1/FLOAT(NPRMN1)
            SX1X2 = SX1X2/FLOAT(NPRMN1)
            SX2X2 = SX2X2/FLOAT(NPRMN1)
C CALCULATE THE COEFFICIENTS
```

```
            DENOM = SX1X1 * SX2X2 - SX1X2 ** 2

            ENN = ( SX1Y * SX2X2 - SX1X2 * SX2Y) /  DENOM

            EMM = ( SX1X1 * SX2Y - SX1Y * SX1X2 ) / DENOM

            LNCEE = MEANY - ENN*MEANX1 - EMM*MEANX2
            CEE = 10.0**(LNCEE)

      ELSEIF (IREGOP .EQ. 3) THEN

            NPR = 0
            MEANY = 0.0
            MEANX1 = 0.0
            SX1Y = 0.0
            SX1X1 = 0.0
            DO 3275 J = 1, NDIV
               RKC = (1.-RDATA(J))*KC
               DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
               DO 3250 K = 1, NP(J)
                  Y = LOG10(DADN(J,K))
     &                    - EMM * LOG10(1.-RDATA(J))
     &                    - PEE * LOG10(DELK(J,K) - DKTH)
     &                    + QUE * LOG10(RKC - DELK(J,K))
                  MEANY = MEANY + Y
                  MEANX1 = MEANX1 + LOG10(DELK(J,K))
 3250          CONTINUE
               NPR = NPR + NP(J)
 3275       CONTINUE
            MEANY = MEANY/FLOAT(NPR)
            MEANX1 = MEANX1/FLOAT(NPR)

C           NOW CALCULATE SY2, SX2, AND SXY

            DO 3350 J= 1, NDIV
               RKC = (1.-RDATA(J))*KC
               DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
               DO 3300 K = 1, NP(J)
                  Y = LOG10(DADN(J,K))
     &                    - EMM * LOG10(1.-RDATA(J))
     &                    - PEE * LOG10(DELK(J,K) - DKTH)
     &                    + QUE * LOG10(RKC - DELK(J,K))

                  DIFFY = Y - MEANY
                  DIFFX1 = LOG10(DELK(J,K)) - MEANX1
                  SX1Y = SX1Y + DIFFX1 * DIFFY
                  SX1X1 = SX1X1 + DIFFX1 * DIFFX1
 3300          CONTINUE
 3350       CONTINUE

            NPRMN1 = NPR - 1

            SX1Y = SX1Y/FLOAT(NPRMN1)
            SX1X1 = SX1X1/FLOAT(NPRMN1)

C CALCULATE THE COEFFICIENTS

            ENN =  SX1Y / SX1X1

            LNCEE = MEANY - ENN * MEANX1
            CEE = 10.0**(LNCEE)

      ELSEIF(IREGOP .EQ. 4) THEN

            NPR = 0
            MEANY = 0.0
            MEANX1 = 0.0
            MEANX2 = 0.0
            MEANX3 = 0.0
            MEANX4 = 0.0
            SX1Y = 0.0
            SX2Y = 0.0
            SX3Y = 0.0
            SX4Y = 0.0
            SX1X1 = 0.0
```

```
      SX1X2 = 0.0
      SX1X3 = 0.0
      SX1X4 = 0.0
      SX2X2 = 0.0
      SX2X3 = 0.0
      SX2X4 = 0.0
      SX3X3 = 0.0
      SX3X4 = 0.0
      SX4X4 = 0.0

      DO 4275 J = 1, NDIV
         RKC = (1.-RDATA(J))*KC
         DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
         DO 4250 K = 1, NP(J)
            MEANY = MEANY + LOG10(DADN(J,K))
            MEANX1 = MEANX1 + LOG10(DELK(J,K))
            MEANX2 = MEANX2 + LOG10(1.-RDATA(J))
            MEANX3 = MEANX3 + LOG10(DELK(J,K) - DKTH)
            MEANX4 = MEANX4 - LOG10(RKC - DELK(J,K))
4250     CONTINUE
         NPR = NPR + NP(J)
4275  CONTINUE
      MEANY = MEANY/FLOAT(NPR)
      MEANX1 = MEANX1/FLOAT(NPR)
      MEANX2 = MEANX2/FLOAT(NPR)
      MEANX3 = MEANX3/FLOAT(NPR)
      MEANX4 = MEANX4/FLOAT(NPR)
C
      NOW CALCULATE SY2, SX2, AND SXY

      DO 4350 J= 1, NDIV
         RKC = (1.-RDATA(J))*KC
         DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
         DO 4300 K = 1, NP(J)
            DIFFY = LOG10(DADN(J,K)) - MEANY
            DIFFX1 = LOG10(DELK(J,K)) - MEANX1
            DIFFX2 = LOG10(1.-RDATA(J)) - MEANX2
            DIFFX3 = LOG10(DELK(J,K) - DKTH) - MEANX3
            DIFFX4 = - LOG10(RKC-DELK(J,K)) - MEANX4
            SX1Y = SX1Y + DIFFX1 * DIFFY
            SX2Y = SX2Y + DIFFX2 * DIFFY
            SX3Y = SX3Y + DIFFX3 * DIFFY
            SX4Y = SX4Y + DIFFX4 * DIFFY
            SX1X1 = SX1X1 + DIFFX1 * DIFFX1
            SX1X2 = SX1X2 + DIFFX1 * DIFFX2
            SX1X3 = SX1X3 + DIFFX1 * DIFFX3
            SX1X4 = SX1X4 + DIFFX1 * DIFFX4
            SX2X2 = SX2X2 + DIFFX2 * DIFFX2
            SX2X3 = SX2X3 + DIFFX2 * DIFFX3
            SX2X4 = SX2X4 + DIFFX2 * DIFFX4
            SX3X3 = SX3X3 + DIFFX3 * DIFFX3
            SX3X4 = SX3X4 + DIFFX3 * DIFFX4
            SX4X4 = SX4X4 + DIFFX4 * DIFFX4
4300     CONTINUE
4350  CONTINUE

      NPRMN1 = NPR - 1

      SX1Y = SX1Y/FLOAT(NPRMN1)
      SX2Y = SX2Y/FLOAT(NPRMN1)
      SX3Y = SX3Y/FLOAT(NPRMN1)
      SX4Y = SX4Y/FLOAT(NPRMN1)
      SX1X1 = SX1X1/FLOAT(NPRMN1)
      SX1X2 = SX1X2/FLOAT(NPRMN1)
      SX1X3 = SX1X3/FLOAT(NPRMN1)
      SX1X4 = SX1X4/FLOAT(NPRMN1)
      SX2X2 = SX2X2/FLOAT(NPRMN1)
      SX2X3 = SX2X3/FLOAT(NPRMN1)
      SX2X4 = SX2X4/FLOAT(NPRMN1)
      SX3X3 = SX3X3/FLOAT(NPRMN1)
      SX3X4 = SX3X4/FLOAT(NPRMN1)
      SX4X4 = SX4X4/FLOAT(NPRMN1)
C CALCULATE THE COEFFICIENTS
```

```
        DENOM = DETER4(SX1X1, SX1X2, SX1X3, SX1X4, SX1X2, SX2X2,
     &                 SX2X3, SX2X4, SX1X3, SX2X3, SX3X3, SX3X4,
     &                 SX1X4, SX2X4, SX3X4, SX4X4)

        ENN = DETER4(SX1Y, SX2Y, SX3Y, SX4Y, SX1X2, SX2X2,
     &                 SX2X3, SX2X4, SX1X3, SX2X3, SX3X3,
     &                 SX3X4, SX1X4, SX2X4, SX3X4, SX4X4) / DENOM

        EMM = DETER4(SX1X1, SX1X2, SX1X3, SX1X4, SX1Y, SX2Y,
     &               SX3Y, SX4Y, SX1X3, SX2X3, SX3X3, SX3X4,
     &               SX1X4, SX2X4, SX3X4, SX4X4) / DENOM

        PEE = DETER4(SX1X1, SX1X2, SX1X3, SX1X4, SX1X2, SX2X2,
     &               SX2X3, SX2X4, SX1Y, SX2Y, SX3Y, SX4Y,
     &               SX1X4, SX2X4, SX3X4, SX4X4) / DENOM

        QUE = DETER4(SX1X1, SX1X2, SX1X3, SX1X4, SX1X2, SX2X2,
     &               SX2X3, SX2X4, SX1X3, SX2X3, SX3X3, SX3X4,
     &               SX1Y, SX2Y, SX3Y, SX4Y) / DENOM

        LNCEE = MEANY - ENN*MEANX1 - EMM*MEANX2
     &          - PEE*MEANX3 - QUE*MEANX4

        CEE = 10.0**(LNCEE)

      ENDIF

C WRITE OUT THE REGRESSED VALUES

      WRITE(3,6300)
      WRITE(3,6400) CEE, ENN, EMM, PEE, QUE

      RETURN
C-------------------- FORMATS -------------------------------------
 6000 FORMAT(////,13X,'MATERIAL INPUT',///,2X,'DESCRIPTION:',2X,A40,//,
     &       2X,'YIELD STRENGTH',18X,F7.0,//,
     &       2X,'CRITICAL S I F',
     &       18X,F7.0,//,2X,'NUMBER OF DIVISIONS',14X,I1,
     &               //,2X,'REGRESSION OPTION',16X,I1,/)

 6050 FORMAT(//,2X,'THRESHOLD MODEL DESCRIPTION',
     &       //,2X,'DKTHo = ',E12.5,
     &        /,2X,'Co = ',E12.5,
     &        /,2X,'d = ',E12.5)
 6150 FORMAT(//,2X,'STRESS RATIO R = ',F7.2,//,6X,'da/dN',8X,'DELK')

 6200 FORMAT(2X,E12.5,2X,E12.5)

 6300 FORMAT(///,25X,'REGRESSION OUTCOME',///,3X,
     &       ' C  ',12X,' n   ',12X,' m   ',12X,' p   ',12X,' q   ',/)

 6400 FORMAT(E12.5,4(4X,E12.5))

 6600 FORMAT(10X,I1,'-',I1,5X,E12.5,5X,E12.5)
      END
C*************************************************************************
C  FUNCTION DETER4 CALCULATES DETERMINANT OF A 4x4 MATRIX
C
C  PROGRAMMER:  S. SUTHARSHANA
C        DATE:  25 SEP 1989
C     VERSION:  92.1, 92.2, 92.3, 92.4, 92.5
C*************************************************************************

      REAL FUNCTION DETER4 (A11, A21, A31, A41, A12, A22, A32, A42,
     &                A13, A23, A33, A43, A14, A24, A34, A44)

C       IMPLICIT NONE

      REAL A11, A21, A31, A41, A12, A22, A32, A42, A13, A23, A33,
     &       A43, A14, A24, A34, A44

      DETER4 =   A11*(   A22 * (A33*A44 - A34*A43)
```

```
     &                        - A23 * (A32*A44 - A34*A42)
     &                        + A24 * (A32*A43 - A33*A42) )
     &          - A12*(          A21 * (A33*A44 - A34*A43)
     &                        - A23 * (A31*A44 - A34*A41)
     &                        + A24 * (A31*A43 - A33*A41) )
     &          + A13*(          A21 * (A32*A44 - A34*A42)
     &                        - A22 * (A31*A44 - A34*A41)
     &                        + A24 * (A31*A42 - A32*A41) )
     &          - A14*(          A21 * (A32*A43 - A33*A42)
     &                        - A22 * (A31*A43 - A33*A41)
     &                        + A23 * (A31*A42 - A32*A41) )
      RETURN
      END
C***********************************************************************
C SUBROUTINE INPUT READS IN THE DATA AND ECHOES IT
C
C PROGRAMMER : S. SUTHARSHANA
C DATE : DECEMBER 1992
C VERSION : 92.5
C
C  Copyright (C) 1991, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************
      SUBROUTINE INPUT (ANGLE, BLFPER, COEXP, E, EM,
     &                  LOCAT, NBLIFE, NHYPER, NLIFE, NLIFET,
     &                  NRAN, NU, NUMSEG, PERIOD, RSO,
     &                  SE, STRHIS, TRUNC)
C
      IMPLICIT NONE

      INTEGER  MAXBLF, MAXLD, MAXLIF, MAXM, MAXSEG

      REAL PI

      PARAMETER (MAXBLF = 10,  MAXLD = 16, MAXLIF = 1000,
     &           MAXM = 20000, MAXSEG = 10, PI = 3.141592654)

      INTEGER INEUB, IRET, KGROW, KPROB

      INTEGER  FILNUM(MAXLD), I, IOUT, J, LOCAT,
     &         NBLIFE, NHYPER, NLIFE, NLIFET, NLOAD, NRAN,
     &         NUMSEG, TYPE(MAXLD)

      DOUBLE PRECISION  RAND

      REAL     AERDA, AERDB, AERSA, AERSB, AIA, AIB,
     &         AIR1, AIR2, AIT1, AIT2, AIR,
     &         AIT, ANGLE, BLFPER(MAXBLF),
     &         COEXP, AOCA, AOCB, AOCR1, AOCR2, AOCT1, AOCT2,
     &         AOCR, AOCT, ASTRA, ASTRB, DPCMU, DPCSIG,
     &         DSTRA, DSTRB, DTIMU, DTISIG,
     &         DTOMU, DTOSIG, E(MAXSEG), EM, FK(10),
     &         INDIAA, INDIAB, INDIR, INDIR1,
     &         INDIR2, INDIT, INDIT1, INDIT2,
     &         KLAMA, KLAMB, LAMKHA, LAMKHB, LAMKCA, LAMKCB,
     &         LAMGRA, LAMGRB,
     &         LAMNA, LAMNB, LAMNC, LAMND, LAMNMU, LAMNSG,
     &         LAMSA, LAMSB, LAMSC, LAMSD, LAMSMU, LAMSSG,
     &         LAMWA, LAMWB

      REAL     M(2, MAXLD), MSTAT(2), MVARA,
     &         MVARB, NEUBA, NEUBB, NU,
     &         P(MAXLD), PCMU, PCMUA, PCMUB, PCO, PCSIG, PCSIGA,
     &         PCSIGB, PERIOD,
     &         PSTAT, RSO, RT(10), SE(MAXSEG), SSTRA,
     &         SSTRB, STRHIS(MAXLD, MAXM), SX(MAXLD), SXST,
     &         SXY(MAXLD), SXYST, SXZ(MAXLD), SXZST, SY(MAXLD),
     &         SYST, SYZ(MAXLD), SYZST, SZ(MAXLD), SZST,
     &         T(MAXLD), THICA, THICB, THICR, THICR1,
     &         THICR2, THICT, THICT1, THICT2, TIMU, TIMUA,
     &         TIMUB, TISIG, TISIGA, TISIGB, TOMU,
     &         TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
     &         TRUNC, TSTAT, V(2, MAXLD),
```

```
&          VSTAT(2), WITHA, WITHB,
&          WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
&          WOFFA, WOFFB, WOFFC, WOFFD,
&          WOFFE, WOFFHI, WOFFLO, WOFFR, WOFFR1, WOFFR2, WOFFR3,
&          WOFFR4, WOFFT, WOFFT1, WOFFT2, WOFFT3, WOFFT4

 CHARACTER*6 LDNAME(MAXLD)

 COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
&              P, T, M, V, PCO, SXST, SYST, SZST, SXYST,
&              SXZST, SYZST, SX, SY, SZ, SXY, SXZ, SYZ

 COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AIA,
&    AIB, AIR1, AIR2, AIT1, AIT2, AIR, AIT,
&    AOCA, AOCB, AOCR1, AOCR2, AOCT1, AOCT2, AOCR, AOCT,
&    ASTRA, ASTRB,
&    DPCMU, DPCSIG, DSTRA, DSTRB, DTIMU, DTISIG, DTOMU, DTOSIG,
&    INDIAA,INDIAB,INDIR1, INDIR2, INDIR, INDIT1, INDIT2, INDIT,
&    KLAMA, KLAMB, LAMGRA, LAMGRB, LAMKHA, LAMKHB, LAMKCA, LAMKCB,
&    LAMNA, LAMNB, LAMNC,  LAMND, LAMNMU, LAMNSG,
&    LAMSA, LAMSB, LAMSC,  LAMSD, LAMSMU, LAMSSG,
&    LAMWA, LAMWB, MVARA, MVARB, NEUBA,  NEUBB,
&    PCMU, PCMUA, PCMUB, PCSIG, PCSIGA, PCSIGB,
&    RAND,
&    SSTRA, SSTRB,
&    THICA, THICB, THICR1, THICR2, THICR, THICT1, THICT2, THICT,
&    TIMU, TIMUA, TIMUB, TISIG, TISIGA, TISIGB,
&    TOMU, TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
&    WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
&    WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
&    WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFR, WOFFT1, WOFFT2,
&    WOFFT3, WOFFT4, WOFFT

 COMMON/FKVSRT/FK, RT
 COMMON/NAMES/LDNAME

 COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
 COMMON  IOUT

 LOGICAL FTEST

 DATA (FILNUM(I), I = 1, MAXLD) /
&     11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
&     24, 25, 26/

 READ(1,*) KPROB
 WRITE(8,*)'PROBLEM TYPE (HEX COIL = 1, EXHEX = 2) =', KPROB
 IF(KPROB .LT. 1 .OR. KPROB.GT. 2) THEN
    WRITE(8,*) 'INVALID PROBLEM TYPE SPECIFICATION'
    CALL TRMNAT
 ENDIF
 READ(1,*) KGROW
 WRITE(8,*)'FORMAN EQUATION WITH m (CONST = 1, VARY = 2) =', KGROW
 IF(KGROW .LT. 1 .OR. KGROW .GT. 2) THEN
    WRITE(8,*) 'INVALID FORMAN EQUATION SPECIFICATION'
    CALL TRMNAT
 ENDIF
 READ(1,*) RAND
 WRITE(8,*)'                          RANDOM NUMBER SEED =', RAND
 READ(1,*) IOUT
 WRITE(8,*)'IOUT - OUTPUT CONTROL VARIABLE               =',IOUT
 READ(1,*) NLIFE
 WRITE(8,*)'                           INNER LOOP SIZE  =', NLIFE
 READ(1,*) NHYPER
 WRITE(8,*)'                           OUTER LOOP SIZE  =',NHYPER
 READ(1,*) IRET
 WRITE(8,*)'      RETARDATION SWITCH  (0 - NO, 1 - YES) =', IRET
 IF(IRET .LT. 0 .OR. IRET .GT. 1) THEN
    WRITE(8,*) 'INVALID RETARDATION SWITCH SPECIFICATION'
    CALL TRMNAT
 ENDIF
 READ(1,*) INEUB
 WRITE(8,*)'      NEUBER SWITCH (0 - NO, 1 - YES)       =', INEUB
 IF(INEUB .LT. 0 .OR. INEUB .GT. 1) THEN
    WRITE(8,*) 'INVALID NEUBERS RULE SPECIFICATION'
```

```
         CALL TRMNAT
      ENDIF

C       CALCULATE TOTAL NUMBER OF LIVES ... IF NLIFET = 1 DETERMINISTIC RUN

      NLIFET = NLIFE * NHYPER

      READ(1,*) NBLIFE
      IF(NBLIFE .GT. 0) THEN
         READ(1,*) (BLFPER(J), J =1, NBLIFE)
      ENDIF

C  READ DRIVER INFORMATION

      IF (KPROB .EQ. 1) THEN
         READ(1,*) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2,
     &             WOFFC, WOFFD, WOFFR3, WOFFR4, WOFFT3, WOFFT4,
     &             WOFFE,
     &             INDIAA,INDIAB,INDIR1, INDIR2, INDIT1, INDIT2,
     &             THICA, THICB, THICR1, THICR2, THICT1, THICT2,
     &             AOCA, AOCB, AOCR1, AOCR2, AOCT1, AOCT2
      ELSEIF (KPROB .EQ. 2) THEN
         READ(1,*) WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2
      ENDIF

      READ(1,*) AIA, AIB, AIR1, AIR2, AIT1, AIT2,
     &          LAMNA, LAMNB, LAMNC,  LAMND,
     &          LAMSA, LAMSB, LAMSC,  LAMSD

      IF (KPROB .EQ. 1) THEN
         READ(1,*) TIMUA, TIMUB, TISIGA, TISIGB,
     &             TOMUA, TOMUB, TOSIGA, TOSIGB,
     &             PCMUA, PCMUB, PCSIGA, PCSIGB

C CALCULATE SOME DRIVER VARIABLES

         DTIMU = TIMUB - TIMUA
         DTISIG = TISIGB - TISIGA
         DTOMU = TOMUB - TOMUA
         DTOSIG = TOSIGB - TOSIGA
         DPCMU = PCMUB - PCMUA
         DPCSIG = PCSIGB - PCSIGA

         IF (IOUT .EQ. 15) THEN
            WRITE(8,*) 'DTIMU = ', DTIMU, ' DTISIG = ', DTISIG
            WRITE(8,*) 'DTOMU = ', DTOMU, ' DTOSIG = ', DTOSIG
            WRITE(8,*) 'DPCMU = ', DPCMU, ' DPCSIG = ', DPCSIG
         ENDIF

C READ ACCURACY FACTORS

         READ(1,*) LAMWA, LAMWB, AERDA, AERDB, AERSA, AERSB,
     &             ASTRA, ASTRB, DSTRA, DSTRB
         IF(INEUB .EQ. 1) THEN
            READ(1,*) NEUBA, NEUBB
         ENDIF
      ELSE
         READ(1,*) SSTRA, SSTRB, DSTRA, DSTRB
      ENDIF


      READ(1,*) LAMKHA, LAMKHB, LAMKCA, LAMKCB,
     &          KLAMA, KLAMB, LAMGRA, LAMGRB

      IF(KGROW .EQ. 2) THEN
         READ(1,*)  MVARA, MVARB
      ENDIF

C READ THE LOADS OR STRESSES

      IF(KPROB .EQ. 1) THEN
         READ(1,*) NLOAD, PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1),
     &             VSTAT(2)
         DO 15 I = 1, NLOAD
            READ(1,*) LDNAME(I), TYPE(I), P(I), T(I), M(1,I), M(2,I),
```

```
     &        V(1,I), V(2,I)
           IF ((TYPE(I) .LT. 1) .OR. (TYPE(I) .GT. 3)) THEN
              WRITE(8,*) 'ERROR:  LOAD INCORRECTLY TYPED'
              CALL TRMNAT
           ENDIF
   15   CONTINUE
      ELSEIF(KPROB .EQ. 2) THEN
        READ(1,*) NLOAD, SXST, SYST, SZST, SXYST, SXZST, SYZST
        DO 16 I = 1, NLOAD
           READ(1,*) LDNAME(I), TYPE(I), SX(I), SY(I), SZ(I), SXY(I),
     &              SXZ(I), SYZ(I)
           IF ((TYPE(I) .LT. 1) .OR. (TYPE(I) .GT. 2)) THEN
              WRITE(8,*) 'ERROR:  LOAD INCORRECTLY TYPED'
              CALL TRMNAT
           ENDIF
   16   CONTINUE
      ENDIF

C READ MISCELLANEOUS INFO

      IF(KPROB .EQ. 1) THEN
         READ(1,*) PCO, LOCAT, ANGLE
      ENDIF

      READ(1,*) RSO, PERIOD, TRUNC, NRAN

C ECHO DATA TO CRKRES

      WRITE(3,900)
      IF(KPROB .EQ. 1) THEN
         WRITE(3,901) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2,
     &              WOFFC, WOFFD, WOFFR3, WOFFR4, WOFFT3, WOFFT4,
     &              WOFFE
         WRITE(3,904) INDIAA,INDIAB,INDIR1, INDIR2, INDIT1, INDIT2
         WRITE(3,905) THICA, THICB, THICR1, THICR2, THICT1, THICT2
         WRITE(3,911) AOCA, AOCB, AOCR1, AOCR2, AOCT1, AOCT2
      ELSEIF(KPROB .EQ. 2) THEN
         WRITE(3,902) WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2
      ENDIF
      WRITE(3,910) AIA, AIB, AIR1, AIR2, AIT1, AIT2
      WRITE(3,906) LAMNA, LAMNB, LAMNC,  LAMND
      WRITE(3,907) LAMSA, LAMSB, LAMSC,  LAMSD
      IF (KPROB .EQ. 1) THEN
         WRITE(3,908) TIMUA, TIMUB, TISIGA, TISIGB,
     &              TOMUA, TOMUB, TOSIGA, TOSIGB,
     &              PCMUA, PCMUB, PCSIGA, PCSIGB

         WRITE(3,9081) LAMWA, LAMWB, AERDA, AERDB, AERSA, AERSB,
     &                ASTRA, ASTRB, DSTRA, DSTRB
         IF(INEUB .EQ. 1) THEN
            WRITE(3,9083) NEUBA, NEUBB
         ENDIF
      ELSE
         WRITE(3,9082) SSTRA, SSTRB, DSTRA, DSTRB
      ENDIF


      WRITE(3,909) LAMKHA, LAMKHB, LAMKCA, LAMKCB,
     &             KLAMA, KLAMB, LAMGRA, LAMGRB

      IF(KGROW .EQ. 2) THEN
         WRITE(3,9091) MVARA, MVARB
      ENDIF

      IF(KPROB.EQ.1) THEN
         WRITE(3,920) PSTAT,TSTAT,MSTAT(1),MSTAT(2),VSTAT(1),VSTAT(2)
         DO 20 I = 1, NLOAD
            WRITE(3,921) LDNAME(I),P(I),T(I),M(1,I),M(2,I),V(1,I),
     &                  V(2,I)
   20    CONTINUE
      ELSEIF(KPROB.EQ.2) THEN
         WRITE(3,922) SXST, SYST, SZST, SXYST, SXZST, SYZST
         DO 21 I = 1, NLOAD
            WRITE(3,921) LDNAME(I), SX(I), SY(I), SZ(I), SXY(I),
     &                  SXZ(I), SYZ(I)
```

```
   21      CONTINUE
       ENDIF

       WRITE(3,924)

       IF(KPROB .EQ. 1) THEN
          WRITE(3,925) PCO, LOCAT, ANGLE
       ENDIF

       WRITE(3,926) RSO, PERIOD, TRUNC, NLOAD, NRAN

C  CONVERT ANGLE TO RADIANS FOR CALCULATIONS

       ANGLE = ANGLE/180.00000 * PI

C READ TIME HISTORIES FROM SPECIFIED FILES

       IF (NRAN .GT. MAXM) THEN
          WRITE(8,*) 'ERROR:  STRESS-TIME HISTORY TOO LARGE'
          CALL TRMNAT
       ENDIF

       DO 25 I = 1, NLOAD
          INQUIRE (FILE = LDNAME(I), EXIST = FTEST)
          IF (FTEST .EQV. .TRUE.) THEN
             OPEN (FILNUM(I), FILE = LDNAME(I), STATUS = 'OLD')
             DO 26 J = 1, NRAN
                READ(FILNUM(I),*) STRHIS(I,J)
   26        CONTINUE
             CLOSE (FILNUM(I))
          ELSE
             WRITE(8,*) 'ERROR:  CANNOT OPEN FILE ', LDNAME(I),
     &                  ' DOES NOT EXIST'
             CALL TRMNAT
          ENDIF
   25  CONTINUE

       IF(KPROB .EQ. 1) THEN
          READ(1,*) EM, COEXP, NU
          WRITE(3,927) EM, COEXP, NU

C READ THE Fk VS. Rt CURVE FOR WELD STRESS CONCENTRATION FOR HEX COIL PROBLEM

          WRITE(3,928)
          DO 30 I = 1, 10
             READ(1,*) FK(I), RT(I)
             WRITE(3,929) FK(I), RT(I)
   30     CONTINUE
       ENDIF

C READ IN THE STRESS-STRAIN VALUES IF NEUBER'S RULE IS TO BE USED IN HEX

       IF (KPROB .EQ. 1 .AND. INEUB .EQ. 1) THEN
          READ(1,*) NUMSEG
          WRITE(3,930) NUMSEG
          DO 35 J = 1, NUMSEG
             READ(1,*) SE(J), E(J)
             WRITE(3,931) SE(J), E(J)
   35     CONTINUE
       ENDIF

C===========  FORMAT STATEMENTS TO ECHO INPUT DATA TO CRKRES =========

  900 FORMAT(2X,'Copyright (C) 1991, California Institute of ',
     &       'Technology. U.S. Government',/,2X,'Sponsorship under ',
     &       'NASA Contract NAS7-918 is acknowledged.',////,
     &       30X,'P  R  O  C  R  K',//,33X,'INPUT DATA',///,4X,
     &       'DRIVERS',25X,'PARAMETER DISTRIBUTIONS',//,48X,'RHO',
     &       16X,'THETA')

  901 FORMAT(/,2X,'WELD OFFSET (%)',3X,'Be(',F4.2,',',F5.2,')',6X,
     &       'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')',
     &       /,20X,'Be(',F4.2,',',F5.2,')',6X,'U(',F7.5,',',F8.5,')',
     &       4X,'U(',F4.1,',',F5.1,')',/,20X,'TEST = ',F4.2)
```

```
 902 FORMAT(/,2X,'CHANNEL WIDTH ',4X,'Be(',F6.4,',',F7.4,')',2X,
    &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')

 904 FORMAT(/,2X,'INNER DIAMETER',4X,'Be(',F6.4,',',F7.4,')',2X,
    &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')

 905 FORMAT(/,2X,'WALL THICKNESS',4X,'Be(',F6.4,',',F7.4,')',2X,
    &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')

 906 FORMAT(//,2X,'LAMBDA RANDOM',5X,'k: U(',F7.5,',',F8.5,')',
    &        /,20X,'COEFFICIENT OF VARIATION: ',F5.3,
    &        /,20X,'STRAIN GAGE FACTOR: ',F9.7)

 907 FORMAT(/,2X,'LAMBDA SINE',7X,'k: U(',F7.5,',',F8.5,')',
    &        /,20X,'COEFFICIENT OF VARIATION: ',F5.3,
    &        /,20X,'STRAIN GAGE FACTOR: ',F9.7,/)

 908 FORMAT(/,2X,'INNER TEMPERATURE',4X,'NORMAL:  MU(',
    &        F6.1,',',F7.1,')   SIGMA(',F5.1,',',F6.1,')',
    &        //,2X,'OUTER TEMPERATURE',4X,'NORMAL:  MU(',
    &        F6.1,',',F7.1,')   SIGMA(',F5.1,',',F6.1,')',
    &        //,2X,'INNER PRESSURE',7X,'NORMAL:  MU(',
    &        F6.1,',',F7.1,')   SIGMA(',F5.1,',',F6.1,')',/)

9081 FORMAT(/,2X,'WELD OFFSET K FAC ',3X,'U(',F8.5,',',F9.5,')',
    &       //,2X,'DYN AERO LOAD FAC ',3X,'U(',F8.5,',',F9.5,')',
    &       //,2X,'STAT AERO LOAD FAC',3X,'U(',F8.5,',',F9.5,')',
    &       //,2X,'AERO STR ANAL FAC ',3X,'U(',F8.5,',',F9.5,')',
    &       //,2X,'DYN STR ANAL FAC  ',3X,'U(',F8.5,',',F9.5,')')

9082 FORMAT(/,2X,'STAT STR ANAL FAC ',3X,'U(',F8.5,',',F9.5,')',
    &       //,2X,'DYN STR ANAL FAC  ',3X,'U(',F8.5,',',F9.5,')')

9083 FORMAT(/,2X,'NEUBERS RULE      ',3X,'U(',F8.5,',',F9.5,')')

 909 FORMAT( /,2X,'LAMBDA Kth         ',3X,'U(',F8.5,',',F9.5,')',
    &        //,2X,'LAMBDA Kc          ',3X,'U(',F8.5,',',F9.5,')',
    &        //,2X,'K CALC FAC         ',3X,'U(',F8.5,',',F9.5,')',
    &        //,2X,'GROWTH CALC FAC    ',3X,'U(',F8.5,',',F9.5,')')

9091 FORMAT(/,2X,'GROWTH COEFF m    ',3X,'U(',F8.5,',',F9.5,')')

 910 FORMAT(/,2X,'CRACK SIZE A ',5X,'Be(',F6.4,',',F7.4,')',2X,
    &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')

 911 FORMAT(/,2X,'CRACK SHAPE A/C',3X,'Be(',F6.4,',',F7.4,')',2X,
    &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')

 920 FORMAT(////,28X,'LOADS INPUT',
    &        ///,5X,'P LOADS',5X,'T LOADS',5X,'M2 LOADS',
    &        4X,'M3 LOADS',4X,'V2 LOADS',4X,'V3 LOADS',
    &        //,2X,'STATIC AERO',
    &        /,2X,F9.6,5(3X,E9.3))

 922 FORMAT(////,27X,'STRESS INPUT',
    &        ///,5X,'   SX   ',5X,'   SY   ',5X,'   SZ    ',
    &        4X,'   SXY   ',4X,'   SXZ   ',4X,'   SYZ    ',
    &        //,2X,'STATIC',
    &        /,2X,F9.6,5(3X,E9.3))

 921 FORMAT(2X,A6,/,2X,F9.6,5(3X,E9.3))

 924 FORMAT(////,25X,'MISCELLANEOUS INPUT')

 925 FORMAT(//,2X,'EXTERNAL PRESSURE',31X,F6.0,
    &        //,2X,'ANALYSIS LOCATION',35X,I1,
    &        //,2X,'ANGLE THETA (DEGREES)',28X,F6.1)

 926 FORMAT(/,2X,'WILLENBORG OVERLOAD FACTOR',25X,E12.5,
    &        //,2X,'STRESS-TIME HISTORY PERIOD',25X,F10.5,
    &        //,2X,'STRESS-TIME HISTORY NOISE FILTER',16X,F7.1,
    &        //,2X,'NUMBER OF TIME-VARYING LOADS',23X,I2,
    &        //,2X,'NUMBER OF POINTS IN HISTORIES',19X,I5,/)

 927 FORMAT (//,2X,'ELASTIC MODULUS',32X,E9.3,
```

```
     &          //,2X,'COEFF OF THERMAL EXPANSION',21X,E14.8,
     &          //,2X,'POISSONS RATIO',33X,F5.3)

  928 FORMAT (///,15X,'Fk VS. Rt CURVE INPUT',
     &          //,10X,'Fk',8X,'Rt',/)

  929 FORMAT(5X,F8.2,4X,F8.2,/)

  930 FORMAT (///,25X,'STRESS-STRAIN CURVE INPUT',
     &          //,2X,'MAXIMUM NUMBER OF SEGMENTS ',25X,I1,
     &          //,2X,'STRESS-STRAIN PRODUCT',5X,'STRAIN VALUES',/)

  931 FORMAT(13X,F8.2,10X,F7.5,/)

      RETURN
      END
C**********************************************************************
C  SUBROUTINE SETDEF INITIALIZES THE VARIABLES AND SETS DEFAULT VALUES
C
C  PROGRAMMER : S. SUTHARSHANA
C  DATA : DECEMBER 1992
C  VERSION : 92.5
C**********************************************************************
      SUBROUTINE SETDEF(LIFE, NCRL)

C      IMPLICIT NONE

      INTEGER  MAXLIF

      PARAMETER (MAXLIF = 1000)

      INTEGER  K, NCRL

      REAL     LIFE(MAXLIF)

C INITIALIZE LIFE VARIABLE

      DO 40 K = 1, MAXLIF
         LIFE(K) = 1.0E+36
   40 CONTINUE

C SET THE NUMBER OF CRACK LENGTHS BETWEEN AI AND AF

      NCRL = 25

      RETURN
      END
C**********************************************************************
C  SUBROUTINE STRAN1 PERFORMS THE STRESS TRANSFORMATION FOR THE PARTICULAR
C  MODE AND LOCATION AND CALCULATES EQUIVALENT STRESS HISTORY
C  PROGRAMMER:  S. SUTHARSHANA
C        DATE:  DECEMBER 1992
C     VERSION:  92.5
C**********************************************************************

      SUBROUTINE STRAN1 (AERD, AERS, ASTR, ANGLE, COEXP, DLTAT,
     &                   DSTR, EM, INDIA, LAMN, LAMS, LAMW, LOCAT,
     &                   NRAN, NU, PC, SPR, STRHIS, THIC, WOFF)

C  SUBPROGRAMS: M4L1, M4L2

C      IMPLICIT NONE

      INTEGER  MAXLD, MAXM

      REAL PI

      PARAMETER (MAXLD = 16, MAXM = 20000, PI = 3.141592654)

      COMMON  IOUT

      INTEGER  I, II, IOUT, J, LOCAT, NLOAD, NRAN, TYPE(MAXLD)
```

```
      REAL      AERD, AERS, ASTR, ANGLE, COEXP, DLTAT, DSTR, EM, FK(10),
     &          INDIA, KT(2,2), LAMN, LAMS, LAMW,
     &          M(2, MAXLD), MLAM(2, MAXLD), MSLAM(2), MSTAT(2),
     &          NU, P(MAXLD), PC, PCO, PLAM(MAXLD), PSLAM, PSTAT,
     &          RT(10), SCLFAC, SPR(MAXM), STATIC(4),
     &          STRAMP(4, MAXLD), STRHIS(MAXLD, MAXM), SX(MAXLD), SXST,
     &          SXY(MAXLD), SXYST, SXZ(MAXLD), SXZST, SY(MAXLD),
     &          SYST, SYZ(MAXLD), SYZST, SZ(MAXLD), SZST,
     &          T(MAXLD), THIC, TLAM(MAXLD), TSLAM, TSTAT,
     &          V(2, MAXLD), VLAM(2, MAXLD), VSLAM(2), VSTAT(2), WOFF

      COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
     &             P, T, M, V, PCO,
     &             SXST, SYST, SZST, SXYST, SXZST, SYZST,
     &             SX, SY, SZ, SXY, SXZ, SYZ

      COMMON/FKVSRT/FK, RT

      DATA KT/1.0,1.0,1.0,1.0/

C       SCALE AERO STATIC LOADS

      SCLFAC = AERS * ASTR

      PSLAM = SCLFAC * PSTAT
      TSLAM = 0.0
      MSLAM(1) = SCLFAC * MSTAT(1)
      MSLAM(2) = SCLFAC * MSTAT(2)
      VSLAM(1) = 0.0
      VSLAM(2) = 0.0

C       SCALE TIME-VARYING LOADS

      DO 230 II = 1, NLOAD
          IF (TYPE(II) .EQ. 1) THEN
             SCLFAC = LAMN * DSTR
             PLAM(II) = SCLFAC * P(II)
             TLAM(II) = 0.0
             MLAM(1,II) = SCLFAC * M(1,II)
             MLAM(2,II) = SCLFAC * M(2,II)
             VLAM(1,II) = 0.0
             VLAM(2,II) = 0.0
          ELSE IF (TYPE(II) .EQ. 2) THEN
             SCLFAC = LAMS * DSTR
             PLAM(II) = SCLFAC * P(II)
             TLAM(II) = 0.0
             MLAM(1,II) = SCLFAC * M(1,II)
             MLAM(2,II) = SCLFAC * M(2,II)
             VLAM(1,II) = 0.0
             VLAM(2,II) = 0.0
          ELSE
             SCLFAC = AERD * ASTR
             PLAM(II) = SCLFAC * P(II)
             TLAM(II) = 0.0
             MLAM(1,II) = SCLFAC * M(1,II)
             MLAM(2,II) = SCLFAC * M(2,II)
             VLAM(1,II) = 0.0
             VLAM(2,II) = 0.0
          ENDIF
  230 CONTINUE

      IF (IOUT .EQ. 15) THEN
          WRITE(8,*) 'AERO STATIC LOADS'
          WRITE(8,*) 'P = ', PSLAM, ' T = ', TSLAM,
     &               ' M2 = ', MSLAM(1), ' M3 = ', MSLAM(2),
     &               ' V2 = ', VSLAM(1), ' V3 = ', VSLAM(2)
          WRITE(8,*) 'TIME-VARYING LOADS'
          DO 240  II = 1, NLOAD
             WRITE(8,*) II, ' P = ',PLAM(II), ' T = ',TLAM(II),
     &              ' M2 = ',MLAM(1,II), ' M3 = ',MLAM(2,II),
     &              ' V2 = ',VLAM(1,II), ' V3 = ',VLAM(2,II)
  240     CONTINUE
```

```
      ENDIF

      IF (LOCAT .EQ. 1) THEN

         CALL M4L1 (COEXP, ANGLE, DLTAT, EM, INDIA, KT, LAMW, MLAM,
     &              MSLAM, NLOAD, NU, PLAM, PC, PCO, PSLAM, STATIC,
     &              STRAMP, TLAM, THIC, TSLAM, VLAM, VSLAM, WOFF, FK, RT)

      ELSE IF (LOCAT .EQ. 2) THEN


         CALL M4L2 (COEXP, ANGLE, DLTAT, EM, INDIA, KT, LAMW, MLAM,
     &              MSLAM, NLOAD, NU, PLAM, PC, PCO, PSLAM, STATIC,
     &              STRAMP, TLAM, THIC, TSLAM, VLAM, VSLAM, WOFF, FK, RT)

      ELSE

         WRITE(8,*) 'ERROR:  INVALID LOCATION SPECIFICATION'
         CALL TRMNAT

      ENDIF
C============== DERIVE THE EQUIVALENT STRESS HISTORY =================
      DO 50 J = 1, NRAN
         SPR(J) = STATIC(1)
   50 CONTINUE

      DO 100 I = 1, NLOAD
         DO 150 J = 1, NRAN
            SPR(J) = SPR(J) + STRHIS(I,J) * STRAMP(1,I)
  150    CONTINUE
  100 CONTINUE

      IF (IOUT .EQ. 25) THEN
         DO 125 J = 1, NRAN
            WRITE(8,*) J, 'SPR = ', SPR(J)
  125    CONTINUE
      ENDIF

      RETURN
      END
C***************************************************************************
C  SUBROUTINE STRAN2 PERFORMS THE STRESS CALCULATION FOR THE EXHEX
C  PROGRAMMER:  S. SUTHARSHANA
C        DATE:  19 NOV 1989
C     VERSION:  92.1, 92.2, 92.3, 92.4, 92.5
C***************************************************************************

      SUBROUTINE STRAN2 (DSTR, LAMN, LAMS, NRAN, SPR, SSTR, STRHIS)

C      IMPLICIT NONE

      INTEGER  MAXLD, MAXM

      REAL PI

      PARAMETER (MAXLD = 16, MAXM = 20000, PI = 3.141592654)

      COMMON  IOUT

      INTEGER  II, IOUT, J, NLOAD, NRAN, TYPE(MAXLD)

      REAL     DSTR, LAMN, LAMS, M(2, MAXLD),
     &         MSTAT(2), P(MAXLD), PCO, PSTAT, SPR(MAXM), SSTR,
     &         STRAMP(MAXLD), STRHIS(MAXLD, MAXM), SX(MAXLD), SXST,
     &         SXY(MAXLD), SXYST, SXZ(MAXLD), SXZST, SY(MAXLD),
     &         SYST, SYZ(MAXLD), SYZST, SZ(MAXLD), SZST,
     &         T(MAXLD), TSTAT, V(2, MAXLD), VSTAT(2)

      COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
     &              P, T, M, V, PCO,
```

```
     &              SXST, SYST, SZST, SXYST, SXZST, SYZST,
     &              SX, SY, SZ, SXY, SXZ, SYZ

C       SET UP THE STRESS AMPLITUDES

      DO 50 II = 1, NLOAD
         IF (TYPE(II).EQ.1) THEN
            STRAMP(II) = LAMN * DSTR * SZ(II)
         ENDIF
         IF (TYPE(II).EQ.2) THEN
            STRAMP(II) = LAMS * DSTR * SZ(II)
         ENDIF
 50   CONTINUE

C       ASSIGN STATIC LOADS

      DO 100 J = 1, NRAN
         SPR(J) = SZST * SSTR
 100  CONTINUE

C       SCALE TIME-VARYING LOADS

      DO 300 II = 1, NLOAD
         DO 200 J = 1, NRAN
               SPR(J) = SPR(J) + STRHIS(II,J) * STRAMP(II)
  200    CONTINUE
  300 CONTINUE

      IF (IOUT .EQ. 25) THEN
         DO 425 J = 1, NRAN
            WRITE(8,*) J, 'SPR = ', SPR(J)
  425    CONTINUE
      ENDIF

      RETURN
      END
C***********************************************************************
C  THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  9MAR87
C  SUBPROGRAM:  GAM
C
C  The random variates are generated using the method described in:
C  Johnson, N. L., and Kotz, S., Distribution in Statistics: Continuous
C    Univariate Distributions - 1, Houghton Mifflin Company, 1970,
C    pp. 181-182.
C***********************************************************************

      SUBROUTINE BETAGN (RAND, RHO, THETA, A, B, X)

      COMMON IOUT

      DOUBLE PRECISION RAND

      REAL    A, B, GAM, RHO, THETA, W, X, Y1, Y2

      INTEGER IOUT

      IF (IOUT .EQ. 15) WRITE(8,*) 'RAND =', RAND, ' RHO =', RHO,
     &          ' THETA =', THETA, ' A =', A, ' B =', B, ' X =', X
      Y1 = GAM((RHO * THETA + 1.), RAND)
      Y2 = GAM(((1. - RHO) * THETA + 1.), RAND)
      W = Y1 / (Y1 + Y2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'Y1 =', Y1, ' Y2 =', Y2, ' W =', W

C  TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

      X = W * (B - A) + A
      IF (IOUT .EQ. 15) WRITE(8,*) 'W =', W, ' X =', X

      RETURN
      END
```

```
C***********************************************************************
C  The random variates are generated using an "Acceptance/Rejection Method"
C  Fishman, George S., "Sampling From the Gamma Distribution on a
C    Computer," Communications of the ACM, Volume 19, Number 7, July 1976,
C    pp. 407-409.

      REAL FUNCTION GAM (ALPHA, RAND)
C     SUBPROGRAM:  RANDOM
      COMMON  IOUT
      INTEGER IOUT
      REAL    A, ALPHA, ARG, U1, U2, V1, V2
      DOUBLE PRECISION RAND

      A = ALPHA - 1.
C     IF (IOUT .EQ. 15) WRITE(8,*) 'A =', A, ' ALPHA =', ALPHA

   10 CALL RANDOM (U1, RAND)
      CALL RANDOM (U2, RAND)
      V1 = - ALOG(U1)
      V2 = - ALOG(U2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =',
C    &                      V1, ' V2 =', V2
      ARG = A * (V1 - ALOG(V1) - 1.)
      IF (V2 .LT. ARG) GOTO 10

      GAM = ALPHA * V1
C     IF (IOUT .EQ. 15) WRITE(8,*) 'GAMMA =', GAM

      RETURN
      END
```

Section 7.2

Low Cycle Fatigue Failure Program BLDLCF

The program tree structures, list of subprograms, descriptions of the key variables, and the FORTRAN source listings for the low cycle fatigue analysis code BLDLCF are given here. The pertinent LCF methodology is given in Section 3. The overall description of the program and the flowcharts are given in Section 5.2. The user's guide for running BLDLCF is given in Section 6.2.

7.2.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for BLDLCF, using Uniform variation on the materials shape parameter *m*, is given in Figure 7.2-1, while the tree structure for the truncated Normal case is given in Figure 7.2-2. The tree structure for BLDLCF V3.4B1.3 is given in Figure 7.2-3. In all trees, those subprograms not "shadow-boxed" are part of the materials characterization model. The program, subprogram, and file names are indicated by UPPERCASE letters.

7.2.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7.2-1. The section numbers where the subprograms are described by means of flowcharts are given next to the names.

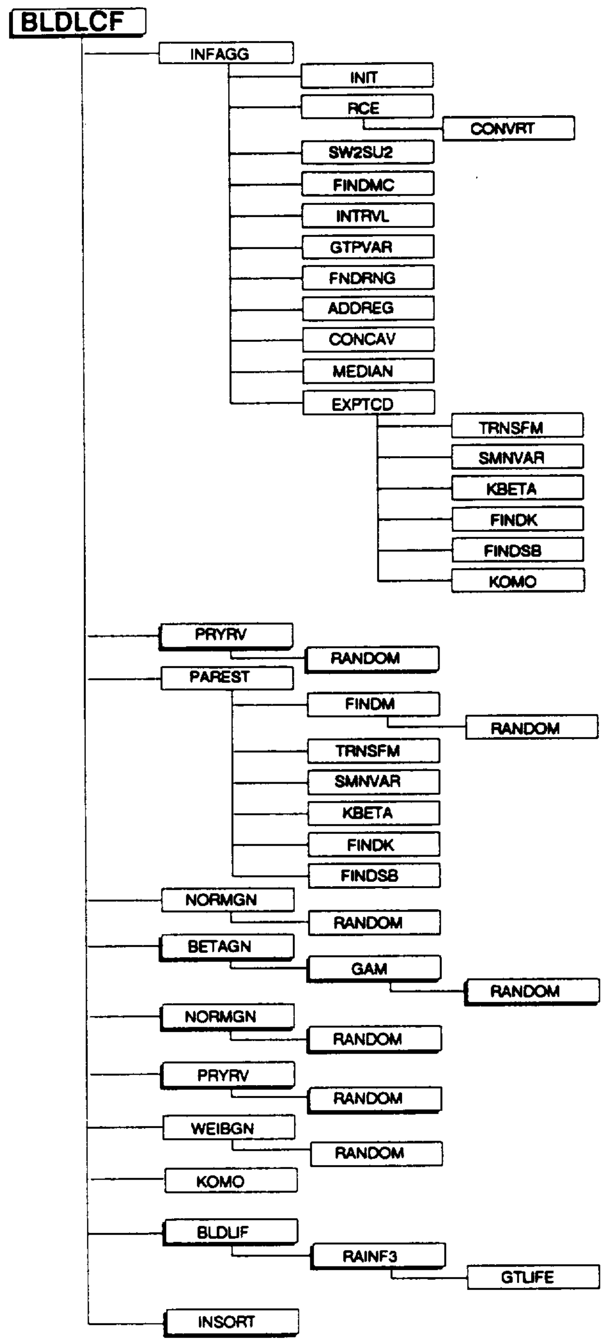


Figure 7.2-1 Tree Structure for Program BLDLCF for the Uniform Variation in Materials Shape Parameter *m*

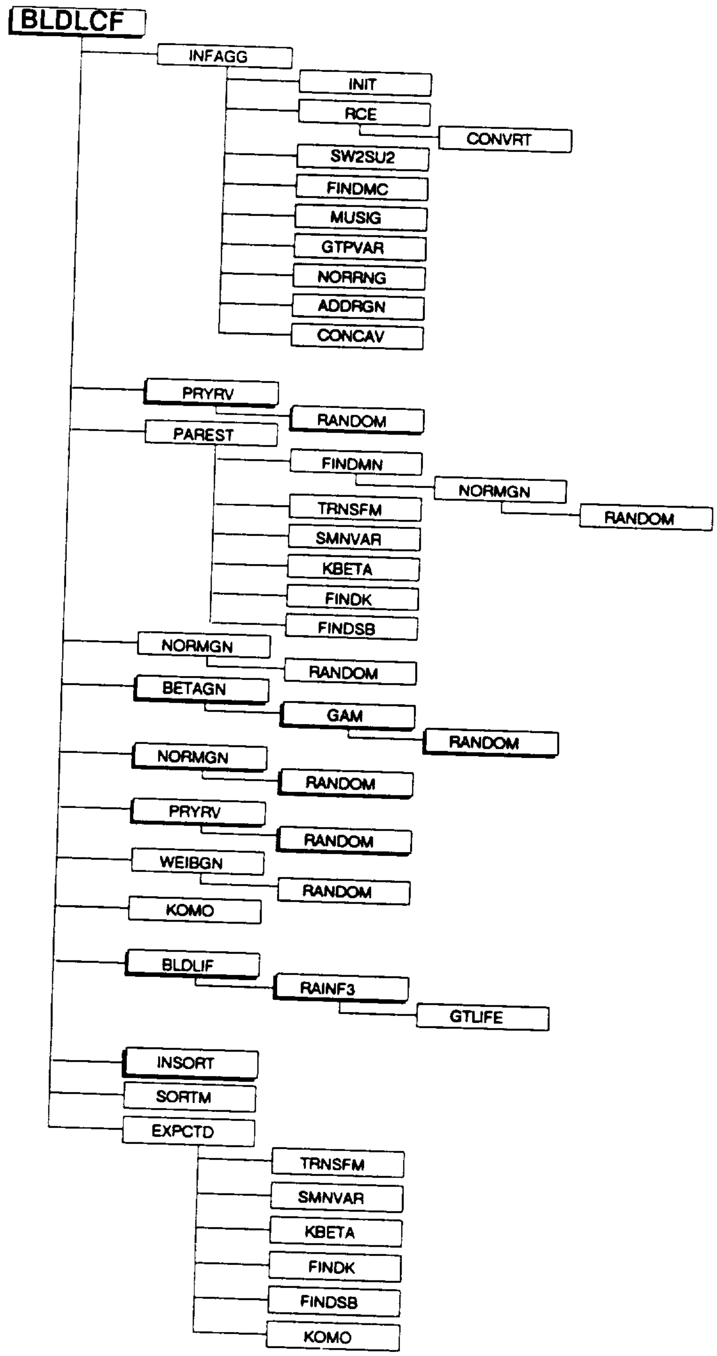


Figure 7.2-2 Tree Structure for Program BLDLCF for the Truncated Normal Variation in Materials Shape Parameter *m*

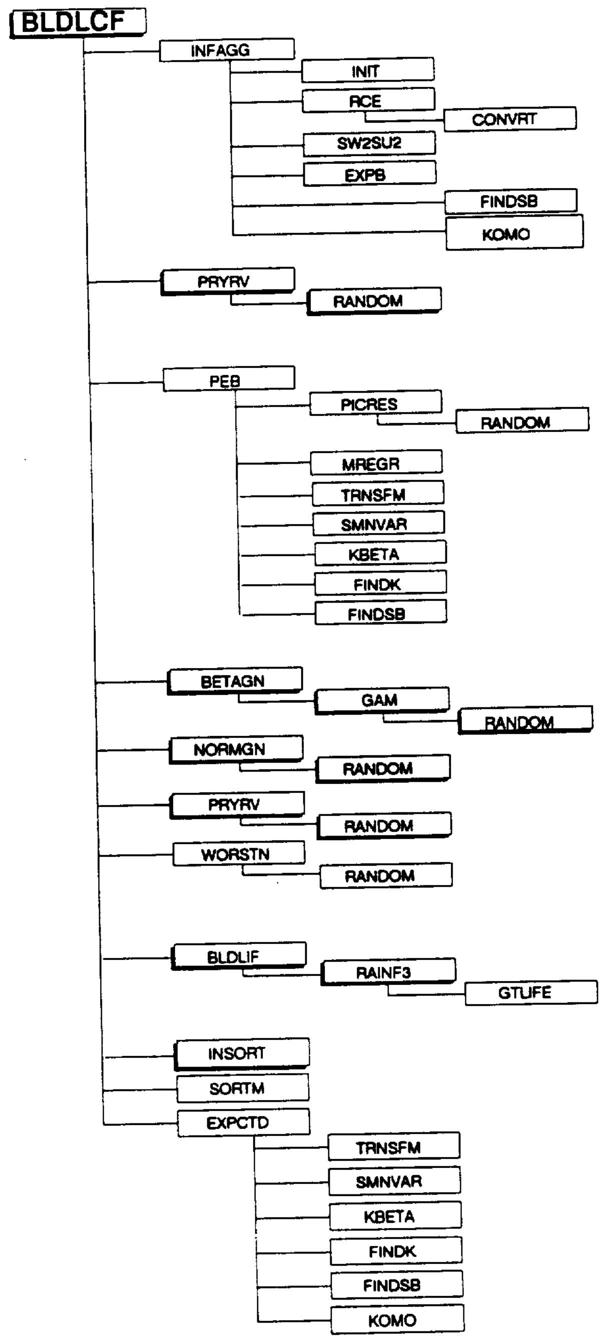


Figure 7.2-3 Tree Structure for Program BLDLCF V3.4B1.3 for the Bootstrapping of the Materials Shape Parameter *m*

Table 7.2-1 List of Subprograms For Program BLDLCF
(Footnotes are at the end of the table)

NAME	SECTION	PURPOSE
ADDREG[1]	4.1.3.9*	Adds the m ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.
ADDRGN[1]	4.1.3.15*	Adds the m ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.
BETAGN[2]	4.4.5*	Generates Beta(a, b, ρ, θ) random variates.
BLDLCF	5.2.2.1	The main routine that controls the logical flow of the low cycle fatigue turbine blade program.
	5.2.3.1	The main routine that controls the logical flow of the low cycle fatigue turbine blade program with the nonparametric materials characterization model.
BLDLIF	5.2.2.2	Performs the calculations of the driver transformation and then calls RAINF3 to calculate the fatigue life.
CONCAV[3]	4.1.3.10*	Adjusts the upper bound of the posterior ranges on m to be consistent with concavity constraints.
CONVRT[4]	4.1.3.3*	Transforms strain data to equivalent zero-mean strains with strain ratio of −1.0.
EXPB	5.2.3.4	Calculates the median S/N curve parameters from the results of the linear regression and residual calculations of Section 3.2.7.
EXPCTD[5]	4.1.3.12*	Calculates the median S/N curve parameters from the results of the information aggregation calculations.
FINDK	4.1.5.6*	Calculates the value of the location parameter K (where $A = K^m$) for each life region by using Equations 2-37 and 2-41 of [1].
FINDM[6]	4.1.5.1*	Obtains the value of m for each life region by adjusting the range (to ensure concavity) and then sampling from the Uniform distribution over the appropriate m range.
FINDMC	4.1.3.5*	Calculates the m range implied by the constraint on the coefficient of variation of fatigue strength, C, for each life region, by using Equations 2-28 through 2-32 of [1].
FINDMN[6]	4.1.5.2*	Obtains the value of m for each life region by sampling from the appropriate truncated Normal distribution on m.
FINDSB	4.1.5.7*	Calculates the life region "tie-points" or strain values which correspond to the "life boundaries," conditional on the randomly selected m for each region. Also calculates K, characterizing the specific material S/N data set, which is a function of β_o and k.
FNDRNG[7]	4.1.3.8*	Combines the 95% confidence interval, J_o, with the implicit and explicit constraints on m, to obtain posterior credibility ranges on m for each life region.

Table 7.2-1 List of Subprograms For Program BLDLCF (Cont'd)

NAME	SECTION	PURPOSE
GAM	4.4.4*	Generates Gamma(α, 1) random variates.
GTLIFE	4.1.8*	Calculates the cycles to failure for a particular strain, based upon the materials characterization model S/N curve of Equation 2-48 of [1].
GTPVAR	4.1.3.7*	Calculates σ^2, the extent of departures from the multiple heat median S/N curve warranted by the information available, by using Equation 2-49 of [1].
INFAGG[8]	5.2.3.2	Controls the logical flow for the information aggregation portion of the materials characterization model.
INIT	4.1.3.1*	Initializes the entries of the arrays used in the information aggregation subroutine, INFAGG, to zero.
INSORT	5.B*	Performs an insertion sort for the lowest fifty percent of the lives calculated.
INTRVL	4.1.3.6*	Calculates the 95% confidence intervals I_0 for C, and J_0 for m, for each region by using Equations 2-24 through 2-26 of [1].
KBETA	4.1.5.5*	Calculates k and β_0 from the sample mean and variance of Z, where Z is a function of strain, life, the life region boundaries, and the m's, by using Equation 2-42 of [1].
KOMO[9]	4.1.6*	Calculates K_0 and m_0 for the zero region, the no data region to the left of the first data region. Extends the S/N curve consistent with the tensile point at S_0.
MEDIAN	4.1.3.11*	Calculates the median values of m, based on the posterior credibility ranges of m, by using Equation 2-34 of [1].
MREGR	5.2.3.7	Performs the regression to obtain the parameter m for the non-parametric materials characterization model.
MUSIG[10]	4.1.3.13*	Calculates the posterior Normal distribution parameters, mean m_* and standard deviation σ_*, for each life region of the S/N curve.
NORMGN[11]	4.4.3*	Generates Normal(μ, σ^2) random variates.
NORRNG[7]	4.1.3.14*	Combines the implicit and explicit constraints on m to obtain the posterior credibility ranges of m for each life region.
PAREST[12]	4.1.5*	Controls the logical flow for the parameter estimation model portion of the materials characterization model.
PEB	5.2.3.5	Controls the logical flow of the bootstrapping portion of the non-parametric materials characterization model described in Section 3.2.7.
PICRES	5.2.3.6	Bootstraps the residuals and performs the pseudo S/N data generation described in Section 3.2.7.

Table 7.2-1 List of Subprograms For Program BLDLCF (Cont'd)

NAME	SECTION	PURPOSE
PRYRV[13]	7.6.6*	Generates the Uniform(a, b) and Uniform(c, d) pair of independent random variates.
RAINF3	5.2.2.3	Performs rainflow cycle counting, Miner's rule damage accumulation, and calls GTLIFE to calculate the fatigue life.
RANDOM[13]	4.4.2*	Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.
RCE	4.1.3.2*	Reads the data from BLDLCD and RELATD; calls CONVRT to transform the strain data to a strain ratio of –1.0; and echoes the data to BLDLCO and RELATO. RCE also breaks S/N data sets into regions as specified by the user.
SMNVAR	4.1.5.4*	Calculates the sample mean and variance of Z, where Z is a function of strain, life, the life region boundaries, and the m's, by using Equation 2-42 of [1].
SORTM[14]	4.1.10*	Sorts the m values in increasing order for each life region for the truncated Normal distribution case.
SW2SU2	5.2.3.3	Calculates the residual variances from the Y on X and X on Y regressions for each life region where $Y = \ln(\textit{Endurance cycles})$ and $X = \ln(\textit{Strain})$ by using Equations 2-20 and 2-21 of [1]; to be used in the credibility range calculations.
TRMNAT	4.1.11*	Performs premature program termination when required.
TRNSFM[15]	4.1.5.3*	Performs the calculations necessary to transform the specific material S/N data into the variable Z, where Z is a function of strain, life, the life region boundaries, and the m's.
WEIBGN	4.4.6*	Generates Weibull(β, $\eta(\beta)$) random variates.
WORSTN	5.2.3.8	Performs the "worst of N" selection described in Section 3.2.7.3 for both Weibull and Lognormal distributions.

* See [1].

1 No data regions to the right are discussed in [1], Page 2-17.

2 The Beta distribution is discussed in [1], Page 2-25.

3 Concavity constraints are discussed in [1], Pages 2-13 through 2-14.

4 The strain transformation is discussed in [1], Page 2-7.

5 The median S/N curve parameter estimation calculations are described in [1], Pages 2-15 through 2-18.

6 Selection of the $\{m_j\}$ parameters is discussed in [1], Page 2-15.

7 Combining information to obtain the posterior credibility ranges on m is discussed in [1], Page 2-13.

8 The information aggregation calculations are discussed in [1], Pages 2-6 through 2-14.

9 Extension of the S/N curve to the left is discussed in [1], Page 2-17.

10 Calculation of the truncated Normal distribution parameters is discussed in [1], Page 2-14.

11 The Normal distribution is discussed in [1], Page 2-23.

12 The parameter estimation calculations are discussed in [1], Pages 2-15 through 2-18.

13 The Uniform distribution is discussed in [1], Page 2-23.

14 The need for saving m's is discussed in [1], Page 2-15.

15 The S/N data transformation is discussed in [1], Page 2-16.

7.2.3 Description of Variables

A list of variables used in the ATD-HPFTP first stage turbine blade LCF code, BLDLCF, is given in Table 7.2-2. The variable names are indicated by **BOLD UPPERCASE** letters; the variable "type" can be interpreted as follows: INT is a standard integer variable; RE is a standard real variable; and DRE is a double precision variable. The various array dimensions are defined by using the following parameters: **MAXBLF**, **MAXDAT**, **MAXLIF**, **MAXM**, **MAXMM**, and **MAXREG**.

Table 7.2-2 List of Variables For Program BLDLCF
(Footnotes are at the end of the table)

VARIABLE NAME	TYPE	DESCRIPTION
ALLM(MAXMM, MAXREG)	RE	2-D array containing the materials model shape parameters (m's) for each life region which are to be used in the truncated Normal median S/N curve calculation.[1]
BIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter K, where $A = K^m$, given in Equation 2-12 of [1].
BIGK1	RE	Dummy variable used during calls to subroutine EXPCTD, equal to **BIGK(1)**.
BLDLIF	RE	Real function that performs the calculations of the driver transformation, calls RAINF3 to calculate a fatigue life, and returns the fatigue life (missions).
BLFPER(MAXBLF)	RE	1-D array containing user specified B-lives which are obtained from the simulated failure distribution. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent: e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.
BLFPOS(MAXBLF)	INT	1-D array containing the indices for the array variable **LIFE( )** corresponding to the user-requested simulated failure distribution B-lives contained in variable **BLFPER( )**.
BZERO	RE	Estimate of Weibull distribution shape parameter β_o, that characterizes the intrinsic variation of the S/N data set, by using Equation 2-11 of [1].
DUM	RE	Dummy variable.
EBEND	RE	The randomly selected value for ε_B, the bending strain due to gas bending and blade tilt, given in Equation 3-1.

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
EBENDA	RE	Uniform distribution lower bound of ε_B.
EBENDB	RE	Uniform distribution upper bound of ε_B.
EM(MAXM)	RE	1-D array containing the total mechanical strain-time history, $\varepsilon_M(t_i)$ (%), in Equation 3-1.
EMNOM	RE	ε_{Mnom} (%) in Equation 3-5, the nominal mechanical strain.
EPSL	RE	ε in Equation 3-9, the material's intrinsic variation or scatter, given by a Lognormal random variate.
EPSW	RE	ε in Equation 3-9, the material's intrinsic variation or scatter, given by a Weibull random variate.
ETH(MAXM)	RE	1-D array containing the total thermal strain-time history $\varepsilon_{TH(t_i)}$ (%) in Equation 3-1.
ETHNOM(MAXM)	RE	$\varepsilon_{THnom}(t_i)$ (%) in Equation 3-4, the 1-D array containing the nominal thermal strain-time history.
ETOT(MAXM)	RE	1-D array containing the total strain-time history, $\varepsilon_T(t_i)$ (%), in Equation 3-1.
FA	RE	$f_A(T_{gas}, h_{gas}) + e_A$ in Equation 3-2, the acceleration response surface.
FAA, FAB, FAC, FAD, FAE, FAF	RE	The coefficients for the acceleration response surface $f_A(T_{gas}, h_{gas})$ in Equation 3-2.
FACTR	RE	Equal to FACTOR = **PHI * KRATIO * Z**. Used by the materials model.
FAERRM	RE	Mean, μ, of Normally distributed e_A, the additive modeling uncertainty for the acceleration response surface, given in Equation 3-2.
FAERRS	RE	Standard deviation, σ, of Normally distributed e_A, the additive modeling uncertainty for the acceleration response surface, given in Equation 3-2.
FD1	RE	$f_{D1}(m, T_s) + e_D$ in Equation 3-3, the deceleration response surface for the thermal strain.
FD1A, FD1B, FD1C, FD1D, FD1E, FD1F	RE	The coefficients for the deceleration response surface $f_{D1}(m, T_s)$ in Equation 3-3.
FD2	RE	$f_{D2}(m, T_s)$ in Equation 3-6, the deceleration response surface for the time of deceleration t_d.
FD2A, FD2B	RE	The coefficients for the deceleration response surface $f_{D2}(m, T_s)$ in Equation 3-6.

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
FD3	RE	$f_{D3}(t_d)$ in Equation 3-7, the deceleration response surface for the rotor speed $\omega(t_6)$.
FD3A, FD3B	RE	The coefficients for the deceleration response surface $f_{D3}(t_d)$ in Equation 3-7.
FDERRM	RE	Mean, μ, of Normally distributed e_D, the additive modeling uncertainty for the deceleration response surface, given in Equation 3-3.
FDERRS	RE	Standard deviation, σ, of Normally distributed e_D, the additive modeling uncertainty for the deceleration response surface, given in Equation 3-3.
FIFTY	RE	Variable used to access the fifty-percent point in the **LIFE( )** array.
FTU	RE	Material ultimate strength (%).
FTY	RE	Material yield strength (%).
GTLIFE	RE	Function given by Equation 2-48 of [1] that calculates the fatigue cycles to failure at a given strain.
HGAS	RE	h_{gas} in Equation 3-2, the randomly selected gas film coefficient.
HGASA	RE	Lower bound of the Beta distribution on h_{gas}.
HGASB	RE	Upper bound of the Beta distribution on h_{gas}.
HGASR	RE	Randomly selected Beta distribution location parameter ρ for h_{gas}.
HGASR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution of h_{gas}.
HGASR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution of h_{gas}.
HGAST	RE	Randomly selected Beta distribution shape parameter θ for h_{gas}.
HGAST1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution of h_{gas}.
HGAST2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution of h_{gas}.
I	INT	Controls inner DO loop.
I	INT	Controls DO loop for each point in the time history.

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
IOUT	INT	Output dump controller. **IOUT** = 0, no intermediate calculation output; **IOUT** = 10, materials characterization model calculations; **IOUT** = 15, driver sampling and driver transformation calculations; and **IOUT** = 20, rainflow cycle counting and damage accumulation.
J	INT	Controls DO loop for each B-life.[2]
K	INT	Controls outer DO loop.
KRATIO	RE	Ratio of *MED K*/MED K* in Equation 2-48 of [1]. **KRATIO** is constant over life regions for the materials model.
L	INT	Controls DO loop for each life region of the S/N curve.
LAMA	RE	λ_α in Equation 3-4, the randomly selected uncertainty factor for the coefficient of thermal expansion.
LAMAA	RE	Uniform distribution lower bound of λ_α.
LAMAB	RE	Uniform distribution upper bound of λ_α.
LAMDA	RE	λ_{dam} in Equation 2-91 of [1], the randomly selected damage accumulation model accuracy factor. See [1], Section 2.2.1.4, for a discussion of the damage calculations.
LAMDAA	RE	Uniform distribution lower bound of the damage accumulation model accuracy factor.
LAMDAB	RE	Uniform distribution upper bound of the damage accumulation model accuracy factor.
LAMG	RE	λ_G in Equation 3-4, the randomly selected thermal strain uncertainty factor due to gas temperature variation during start.
LAMGA	RE	Lower bound of the Beta distribution on λ_G.
LAMGB	RE	Upper bound of the Beta distribution on λ_G.
LAMGR	RE	Randomly selected Beta distribution location parameter ρ for λ_G.
LAMGR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution of λ_G.
LAMGR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution of λ_G.

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LAMGT	RE	Randomly selected Beta distribution shape parameter θ for λ_G.
LAMGT1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution of λ_G.
LAMGT2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution of λ_G.
LAMP	RE	λ_P in Equation 3-5, the randomly selected deviation in blade pull load due to uncertainty in blade mass.
LAMPA	RE	Uniform distribution lower bound of λ_P.
LAMPB	RE	Uniform distribution upper bound of λ_P.
LAMTM	RE	λ_{TMF} in Section 3.2.6, the randomly selected thermal-mechanical fatigue (TMF) model accuracy factor.
LAMTMA	RE	Uniform distribution lower bound of the TMF model accuracy factor.
LAMTMB	RE	Uniform distribution upper bound of the TMF model accuracy factor.
LIFE(MAXLIF)	RE	1-D array containing values of the lives generated by program BLDLCF. The lives are sorted values for the left-hand tail simulated failure distribution.
LIFEL(MAXLIF)	RE	1-D array containing values of the lives generated by program BLDLCF V3.4B1.3 for Lognormal intrinsic materials variation. The lives are sorted values for the left-hand tail simulated failure distribution.
LIFEW(MAXLIF)	RE	1-D array containing values of the lives generated by program BLDLCF V3.4B1.3 for Weibull intrinsic materials variation. The lives are sorted values for the left-hand tail simulated failure distribution.
LIFL	RE	Fatigue life value (missions) equal to **EPSL** * **NEWLIF** to be inserted in **LIFEL( )** for the non-parametric materials characterization model with Lognormal intrinsic materials variation.
LIFW	RE	Fatigue life value (missions) equal to **EPSW** * **NEWLIF** to be inserted in **LIFEW( )** for the non-parametric materials characterization model with Weibull intrinsic materials variation.
LNA(0:MAXREG)	RE	1-D array containing values of ln(A) = ln(**BIGK**) * **MM** for each life region of the S/N curve.

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LNPHI	RE	The natural logarithm of φ in Equation 2-11 of [1], the material's intrinsic variation, or scatter, given by a Lognormal(0, **PHISIG**^{2}) random variate.
LNZ	RE	ln(Z) in Equation 2-48 of [1], the Normal(0, **PVAR**) random variate for the materials process variation aspect of the materials model.
LPHIM(0:MAXREG)	RE	1-D array containing values of ln(PHI) * **MM** for each life region of the S/N curve.
M	INT	Controls symmetry DO loop.
MANAL	RE	The randomly selected mechanical strain analysis accuracy factor, λ_{MA} in Equation 3-5.
MANALA	RE	Uniform distribution lower bound of λ_{MA}.
MANALB	RE	Uniform distribution upper bound of λ_{MA}.
MAXBLF	INT	Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is 10.[2]
MAXDAT	INT	Maximum number of points per data set per region allowed for the S/N curve. The maximum number of data points per set allowed is 50.
MAXLIF	INT	Maximum number of fatigue lives allowed for the simulated failure distribution. The maximum number of fatigue lives to be saved is 10,000.
MAXM	INT	Maximum number of points allowed in the time history arrays. The maximum number of points is 50.
MAXMM	INT	Maximum number of *m*'s to be saved and sorted for the truncated Normal median S/N curve.[1] The maximum number of *m*'s is 20,000.
MAXREG	INT	Maximum number of life regions allowed for the S/N curve. The maximum number of regions is 3.
MCOUNT	INT	Counts number of *m*'s to be used to calculate the median S/N curve for the truncated Normal distribution case.[1]
MEDKB(0:MAXREG)	RE	1-D array containing the median *K* for each life region of the S/N curve for the bootstrapping option.
MEDM(MAXMM)	RE	1-D array containing the empirical median *m* for each life region of the S/N curve.[3]

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
MEDMB(0:MAXREG)	RE	1-D array containing the median *m* for each life region of the S/N curve for the bootstrapping option.
MID	INT	Pointer to the median *m* values in array **SORTM( )** for the truncated Normal median S/N curve. Value of half of **MCOUNT**.
MINPHI	RE	Value of min(**PHI**), the minimum of **NSYM** draws of the materials scatter parameter φ.
MM(0:MAXREG)	RE	m_j in Equation 2-12 of [1], the 1-D array containing randomly selected values of the materials model shape parameter *m* for each life region of the S/N curve.
MODER1	RE	e_A in Equation 3-2, the randomly selected additive modeling uncertainty for the acceleration response surface.
MODER2	RE	e_D in Equation 3-3, the randomly selected additive modeling uncertainty for the deceleration response surface.
MPROC	INT	Materials PROCess variation. Controls materials process variation. A value of 0 indicates no materials process variation, while a value of 1 indicates that materials process variation should be included.[4]
MU(MAXREG)	RE	1-D array containing the posterior Normal distribution mean[5] of the materials shape parameter *m* for each life region of the truncated Normal S/N curve.
NBLIFE	INT	Number of B-lives to be obtained from the simulated failure distribution.[2]
NBND(0:MAXREG)	RE	$N*_{i,\,i+1}$ in Equation 2-35 of [1], the 1-D array containing upper bounds for the **NUMREG** life regions of interest for the specific material S/N data set.
NEWLIF	RE	Fatigue life value (missions) returned from call to function BLDLIF.
NF(MAXDAT, MAXREG)	RE	2-D array containing values from the array **RAWNF( )** for the specific material S/N data set partitioned into life regions.
NHYPER	INT	The outer loop size.
NLIFE	INT	The inner loop size.

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
NLIFET	INT	Total number of lives calculated by program BLDLCF. Value of **NHYPER * NLIFE**.
NMED	INT	Controls S/N curve median calculation for the truncated Normal distribution case. A value of 0 indicates that the user does not desire a median calculation or that the Uniform distribution case is being used; while a value of 1 indicates that the user desires the median calculation to be performed.
NOMSPD	RE	ω_o (rpm) in Equation 3-5, the nominal rotor speed.
NPTS(MAXREG)	INT	1-D array containing the number of points per life region for the specific material S/N data set.
NSYM	INT	Symmetry number, usually equal to the multiplicity of the modeling unit in the component.
NTIME	RE	Number of points in strain-time history.
NUMREG	INT	R in Equation 2-11 of [1], the number of life regions of interest in the S/N curve.
PERIOD	RE	T (missions) in Equation 2-91 of [1], the length of time in missions of the strain-time history.
PHI	RE	φ in Equation 2-11 of [1], the material's intrinsic variation, or scatter, given by a Weibull(β_o, $\eta_o(\beta_o)$) random variate.
PHISIG	RE	σ in the distribution $\Lambda(0, \sigma^2)$ of Section 3.2.7.2, a parameter of the Lognormal distribution of the intrinsic materials variation.
PSIG	RE	σ in Equation 2-48 of [1], the value of SQRT(**PVAR**).
PVAR	RE	σ^2 in Equation 2-48 of [1], characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.
RAINF3	RE	Real function which performs rainflow cycle counting, Miner's Rule damage accumulation, and calls GTLIFE to calculate the fatigue life.
RAND	DRE	Random number seed.
RANGEM(2, MAXREG)	RE	2-D array containing values of the posterior credibility ranges on the materials model shape parameter m for each life region in the S/N curve. **RANGEM(1,L)** is the lower bound and **RANGEM(2,L)** is the upper bound in region L[6]

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
RESID(MAXDAT)	RE	1-D array containing the values of the residuals of the regression for each point in the specific material S/N data for the bootstrapping option.
RESNF(MAXDAT, MAXREG)	RE	1-D array containing values of *N* for the generated pseudo S/N data for the bootstrapping option.
RPM(MAXM)	RE	1-D array containing $\omega(t_i)$ (rpm) in Equation 3-5, the rotor speed time history.
SBND(0:MAXREG)	RE	1-D array containing the strain values (%) with strain ratio = –1.0, corresponding to the "life boundary" values for each life region of the S/N curve contained in array **NBND( )**.
SIG(MAXREG)	RE	1-D array containing the posterior Normal distribution standard deviation[7] of the materials model shape parameter *m* for each life region of the truncated Normal S/N curve.
SLOPE	RE	The randomly selected deceleration slope at shutdown, *m* (°R/sec) in Equation 3-3.
SLOPEA	RE	Lower bound of the Beta distribution on *m*.
SLOPEB	RE	Upper bound of the Beta distribution on *m*.
SLOPR	RE	Randomly selected Beta distribution location parameter ρ for *m*.
SLOPR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution of *m*.
SLOPR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution of *m*.
SLOPT	RE	Randomly selected Beta distribution shape parameter θ for *m*.
SLOPT1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution of *m*.
SLOPT2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution of *m*.
SPEED	RE	$\omega(t_5)$ (rpm) in Equation 3-5, the randomly selected steady state rotor speed.
SPEEDM	RE	Mean, μ, of Normally distributed steady state rotor speed (rpm).
SPEEDS	RE	Standard deviation, σ, of Normally distributed steady state rotor speed (rpm).

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
STR(MAXDAT, MAXREG)	RE	2-D array containing strain points with strain ratio = −1.0, for the specific material S/N data set partitioned into life regions.
SZERO	RE	Strain tensile test point, S_o (%).[8]
TANAL	RE	The randomly selected thermal strain analysis accuracy factor, λ_{TA} in Equation 3-4.
TANALA	RE	Uniform distribution lower bound of λ_{TA}.
TANALB	RE	Uniform distribution upper bound of λ_{TA}.
TGAS	RE	T_{gas} (°R) in Equation 3-2, the randomly selected gas temperature at t_1.
TGASA	RE	Lower bound of the Beta distribution on T_{gas}.
TGASB	RE	Upper bound of the Beta distribution on T_{gas}.
TGASR	RE	Randomly selected Beta distribution location parameter ρ for T_{gas}.
TGASR1	RE	Uniform distribution lower bound of parameter ρ in the Beta distribution of T_{gas}.
TGASR2	RE	Uniform distribution upper bound of parameter ρ in the Beta distribution of T_{gas}.
TGAST	RE	Randomly selected Beta distribution shape parameter θ for T_{gas}.
TGAST1	RE	Uniform distribution lower bound of parameter θ in the Beta distribution of T_{gas}.
TGAST2	RE	Uniform distribution upper bound of parameter θ in the Beta distribution of T_{gas}.
TRBIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter K consistent with the tensile point S_o.[8]
TRSBND(0:MAXREG)	RE	1-D array containing the strain values (%) with strain ratio = −1.0, corresponding to the "life boundary" values for each region of the S/N curve contained in array **NBND( )** for each **PHI** draw consistent with the tensile point S_o.[8]
TRUNC	RE	Value used to filter out noise in the composite strain-time history during rainflow cycle counting. See [1], Section 2.2.1.4, for a discussion of rainflow cycle counting.

Table 7.2-2 List of Variables For Program BLDLCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
TSTART	RE	T_S (°R) in Equation 3-3, the randomly selected gas temperature at the start of deceleration.
TSTMU	RE	Mean, μ, of Normally distributed T_S, the gas temperature at the start of deceleration, given in Equation 3-3.
TSTSIG	RE	Standard deviation, σ, of Normally distributed T_S, gas temperature at the start of deceleration, given in Equation 3-3.
TSUBI	INT	The time index for the rotor time history for which the distribution on steady state rotor speed is valid.
VARPHI	INT	Controls type of material's intrinsic variation desired. A value of 1 indicates Weibull variation and a value of 2 indicates Lognormal variation.
VARY	INT	Controls type of S/N curve variation desired. A value of 0 indicates that no variation is required; a value of 1 means that intrinsic materials variation only is desired; a value of 2 indicates that the user desires a Uniform distribution on m; while a value of 3 indicates that a truncated Normal distribution is desired; a value of 4 indicates the user desires the bootstrapping option.
WEXP	RE	w in Equation 3-8, the exponent for the Walker relation.
Z	RE	Z in Equation 2-48 of [1], the randomly selected process variation shift factor given by a Lognormal(0, **PVAR**) random variate.
ZROREG	INT	ZeRO REGion, the variable permits the inclusion of the tensile point S_o. The value of 0 implies a DO loop from zero to **NUMREG**, while a value of 1 causes the DO loop to be executed from one to **NUMREG**.[8]

1 The need for saving m's is discussed in [1], Page 2-15.

2 See variable **BLFPER( )** for a description of B-life.

3 The median S/N curve for the truncated Normal case is discussed in [1], Page 2-15.

4 See [1], Section 2.1.2.3, for a discussion on process variation in materials.

5 m_* of the posterior density of m is discussed in [1], Page 2-14.

6 The posterior credibility ranges $\pi(m)$ are discussed in [1], Page 2-13.

7 σ_* of the posterior density of m is discussed in [1], Page 2-14.

8 Extension of the S/N curve to the left using the tensile point is discussed in [1], Page 2-17.

7.2.4 Program BLDLCF Listing

BLDLCF Version 3.4

Program BLDLCF Listing Temporal Order, Uniform Distribution

Program BLDLCF Listing Temporal Order, Truncated Normal Distribution

```
C*************************************************************************
C  PROGRAM BLDLCF CONTROLS THE FLOW OF LOGIC OF THE LOW CYCLE
C  FATIGUE ANALYSIS OF THE TURBINE BLADE FOIL PROBLEM
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JAN92     COMMENTS:  3APR92
C     VERSION:  3.4 (MATCHR V8.5, RAINF3 V1.1, INSORT V2.1)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C*************************************************************************

      PROGRAM BLDLCF

C  SUBPROGRAMS:  INFAGG, PAREST, PRYRV, BETAGN, NORMGN, WEIBGN,
C                TRMNAT, BLDLIF, INSORT, SORTM, EXPTCD
C        FILES:  1:BLDLCD-OLD; 3:BLDLCO-NEW; 5:RELATD-OLD; 6:RELATO-NEW;
C                7:DUMP-NEW; 8:IOUTPR-NEW; 9:LOWLIF-NEW;
C                NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

C     IMPLICIT NONE

      INTEGER  MAXBLF, MAXDAT, MAXLIF, MAXM, MAXMM, MAXREG

      PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLIF = 10000,
     &           MAXM = 50, MAXMM = 20001, MAXREG = 3)

      COMMON    IOUT

      INTEGER  BLFPOS(MAXBLF), I, IOUT, J, K, L, M, MCOUNT, MID,
     &         MPROC, NBLIFE, NHYPER, NLIFE, NLIFET, NMED,
     &         NPTS(MAXREG), NSYM, NTIME, NUMREG, TSUBI, VARPHI,
     &         VARY, ZROREG

      DOUBLE PRECISION  RAND

      REAL     ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGK1, BLDLIF,
     &         BLFPER(MAXBLF), BZERO, DUM, EBEND, EBENDA, EBENDB,
     &         EMNOM, ETHNOM(MAXM), FAA, FAB, FAC, FACTR, FAD, FAE,
     &         FAF, FAERRM, FAERRS, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
     &         FD2A, FD2B, FD3A, FD3B, FDERRM, FDERRS, FIFTY, FTU, FTY,
     &         HGAS, HGASA, HGASB, HGASR, HGASR1, HGASR2, HGAST,
     &         HGAST1, HGAST2, KRATIO, LAMA, LAMAA, LAMAB, LAMDA,
     &         LAMDAA, LAMDAB, LAMG, LAMGA, LAMGB, LAMGR, LAMGR1,
     &         LAMGR2, LAMGT, LAMGT1, LAMGT2, LAMP, LAMPA, LAMPB,
     &         LAMTM, LAMTMA, LAMTMB, LIFE(MAXLIF), LNA(0:MAXREG),
     &         LNPHI, LNZ, LPHIM(0:MAXREG), MANAL, MANALA, MANALB,
     &         MEDM(MAXREG), MINPHI, MM(0:MAXREG), MODER1, MODER2,
     &         MU(MAXREG), NBND(0:MAXREG), NEWLIF, NF(MAXDAT, MAXREG),
     &         NOMSPD, PERIOD, PHI, PHISIG, PSIG, PVAR,
     &         RANGEM(2, MAXREG), RPM(MAXM), SBND(0:MAXREG),
     &         SIG(MAXREG), SLOPE, SLOPEA, SLOPEB, SLOPR, SLOPR1,
     &         SLOPR2, SLOPT, SLOPT1, SLOPT2, SPEED, SPEEDM, SPEEDS,
     &         STR(MAXDAT, MAXREG), SZERO, TANAL, TANALA, TANALB, TGAS,
     &         TGASA, TGASB, TGASR, TGASR1, TGASR2, TGAST, TGAST1,
     &         TGAST2, TRBIGK(0:MAXREG), TRSBND(0:MAXREG), TRUNC,
     &         TSTART, TSTMU, TSTSIG, WEXP, Z

C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

      OPEN (1, FILE = 'BLDLCD', STATUS = 'OLD')
      OPEN (3, FILE = 'BLDLCO', STATUS = 'NEW')
      OPEN (7, FILE = 'DUMP',   STATUS = 'NEW')
      OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
      OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')


      READ(1,*) RAND
      WRITE(8,*) '                       RANDOM NUMBER SEED =', RAND
      READ(1,*) IOUT
      WRITE(8,*) 'IOUT (MATCHR = 10, BLDLCF = 15, RAINF3 = 20)=', IOUT
      READ(1,*) NLIFE
      WRITE(8,*) '                          INNER LOOP SIZE =', NLIFE
      READ(1,*) NHYPER
      WRITE(8,*) '                          OUTER LOOP SIZE =',NHYPER
```

```
      READ(1,*) NSYM
      WRITE(8,*) '                              SYMMETRY NUMBER =', NSYM
      READ(1,*) VARY
      WRITE(8,*) '              TYPE OF S/N VARIATION DESIRED '
      WRITE(8,*) ' (0-NONE; 1-INTRINSIC; 2-UNIFORM; 3-NORMAL) =',VARY
      READ(1,*) NMED
      WRITE(8,*) '         NORMAL MEDIAN CURVE (0 - NO, 1 - YES) =',NMED
      READ(1,*) MPROC
      WRITE(8,*) '           MATERIALS PROCESS VARIATION DESIRED'
      WRITE(8,*) '                             (0 - NO, 1 - YES) =', MPROC
      READ(1,*) VARPHI
      WRITE(8,*) '           TYPE OF INTRINSIC VARIATION DESIRED '
      WRITE(8,*) '                  (1 - WEIBULL; 2 - LOGNORMAL) =',VARPHI

      IF ((VARY .LT. 0) .OR. (VARY .GT. 3)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF S/N VARIATION DESIRED'
         CALL TRMNAT
      ENDIF

      IF ((NMED .NE. 0) .AND. (NMED .NE. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID RESPONSE TO NORMAL MEDIAN ',
     &              'CURVE QUESTION'
         CALL TRMNAT
      ENDIF

      IF ((MPROC .LT. 0) .OR. (MPROC .GT. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF MATERIALS PROCESS ',
     &              'VARIATION DESIRED'
         CALL TRMNAT
      ENDIF

      IF ((VARPHI .LT. 1) .OR. (VARPHI .GT. 2)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF INTRINSIC MATERIALS ',
     &              'VARIATION DESIRED'
         CALL TRMNAT
      ENDIF

      READ(1,*) NBLIFE
      IF (NBLIFE .GT. 0) READ(1,*) (BLFPER(J), J = 1, NBLIFE)

C ** READ DATA FROM BLDLCD

      READ(1,*) HGASA,  HGASB,  HGASR1, HGASR2, HGAST1, HGAST2,
     &          TGASA,  TGASB,  TGASR1, TGASR2, TGAST1, TGAST2,
     &          SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
     &          LAMGA,  LAMGB,  LAMGR1, LAMGR2, LAMGT1, LAMGT2,
     &          TSUBI,  SPEEDM, SPEEDS,
     &          FAERRM, FAERRS, TSTMU,  TSTSIG,
     &          FDERRM, FDERRS,
     &          EBENDA, EBENDB, LAMPA,  LAMPB,
     &          MANALA, MANALB, LAMAA,  LAMAB,
     &          TANALA, TANALB, LAMDAA, LAMDAB,
     &          LAMTMA, LAMTMB
      READ(1,*) EMNOM,  NOMSPD, PERIOD, TRUNC, NTIME, WEXP
      READ(1,*) FAA, FAB, FAC, FAD, FAE, FAF,
     &          FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
     &          FD2A, FD2B,
     &          FD3A, FD3B

      IF (NTIME .GT. MAXM) THEN
         WRITE(8,*) 'ERROR:  STRAIN-TIME HISTORY TOO LARGE'
         CALL TRMNAT
      ENDIF

      DO 20 I = 1, (NTIME - 1)
         READ(1,*) RPM(I), ETHNOM(I)
   20 CONTINUE

C ** ECHO DATA TO BLDLCO

      WRITE(3,900)
      WRITE(3,901) HGASA,  HGASB,  HGASR1, HGASR2, HGAST1, HGAST2,
     &             TGASA,  TGASB,  TGASR1, TGASR2, TGAST1, TGAST2
      WRITE(3,902) SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
     &             LAMGA,  LAMGB,  LAMGR1, LAMGR2, LAMGT1, LAMGT2
```

```
      WRITE(3,903) TSUBI,  SPEEDM, SPEEDS, FAERRM, FAERRS,
     &             TSTMU,  TSTSIG, FDERRM, FDERRS
      WRITE(3,904) EBENDA, EBENDB, LAMPA,  LAMPB,  MANALA, MANALB,
     &             LAMAA,  LAMAB,  TANALA, TANALB
      WRITE(3,905) EXP(LAMDAA), EXP(LAMDAB), EXP(LAMTMA), EXP(LAMTMB)
      WRITE(3,906) EMNOM,  NOMSPD, PERIOD, TRUNC, NTIME, WEXP
      WRITE(3,907) FAA, FAB, FAC, FAD, FAE, FAF,
     &             FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
     &             FD2A, FD2B,
     &             FD3A, FD3B

      DO 25 I = 1, (NTIME - 1)
         WRITE(3,908) RPM(I), ETHNOM(I)
   25 CONTINUE

C ** CALL INFAGG TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECT
C       OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS

      CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, ZROREG, NUMREG,
     &             NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR)

      IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)

      MCOUNT = 0

C ** INITIALIZE VARIABLES

      DO 35 K = 1, MAXLIF
         LIFE(K) = 1.0E+36
   35 CONTINUE

      NLIFET = NHYPER * NLIFE

C ** OUTER LOOP - THIS LOOP SAMPLES HYPER-PARAMETER SETS

      DO 150 K = 1, NHYPER

C ** CALL PRYRV TO OBTAIN RHO,THETA PAIRS FOR INNER LOOP CALCULATIONS

         CALL PRYRV (RAND, HGASR1, HGASR2, HGAST1, HGAST2, HGASR, HGAST)
         CALL PRYRV (RAND, TGASR1, TGASR2, TGAST1, TGAST2, TGASR, TGAST)
         CALL PRYRV (RAND, SLOPR1, SLOPR2, SLOPT1, SLOPT2, SLOPR, SLOPT)
         CALL PRYRV (RAND, LAMGR1, LAMGR2, LAMGT1, LAMGT2, LAMGR, LAMGT)
         CALL PRYRV (RAND, MANALA, MANALB, TANALA, TANALB, MANAL, TANAL)

C ** CALL PAREST TO PERFORM THE PARAMETER ESTIMATION ASPECT OF THE
C       MATERIALS CHARACTERIZATION MODEL CALCULATIONS

         CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG,
     &                RAND, NBND, STR, BIGK, BZERO, MM, SBND)

         PHISIG = 1.282550 / BZERO

C ** OBTAIN MATERIALS PROCESS VARIATION IF DESIRED

         CALL NORMGN (RAND, 0.0, PSIG, LNZ)

         IF (MPROC .EQ. 1) THEN
            Z = EXP (LNZ)
         ELSE
            KRATIO = 1.0
            Z = 1.0
            LNZ = 0.0
         ENDIF

         MCOUNT = MCOUNT + 1
         DO 175 L = 1, NUMREG
            ALLM(MCOUNT, L) = MM(L)
  175    CONTINUE

C ** INNER LOOP - THIS LOOP GENERATES BLADE FAILURE TIMES

         DO 200 I = 1, NLIFE

C ** INITILIZE S/N CURVE PARAMETERS
```

```
            DO 225 L = 0, MAXREG
               LNA(L) = 0.0
               LPHIM(L) = 0.0
               TRSBND(L) = 0.0
  225       CONTINUE

C ** SELECT DRIVERS FOR CALCULATING LIFE

            CALL BETAGN (RAND, HGASR, HGAST, HGASA,  HGASB,  HGAS)
            CALL BETAGN (RAND, TGASR, TGAST, TGASA,  TGASB,  TGAS)
            CALL BETAGN (RAND, SLOPR, SLOPT, SLOPEA, SLOPEB, SLOPE)
            CALL BETAGN (RAND, LAMGR, LAMGT, LAMGA,  LAMGB,  LAMG)

            CALL NORMGN (RAND, SPEEDM, SPEEDS, SPEED)
            CALL NORMGN (RAND, FAERRM, FAERRS, MODER1)
            CALL NORMGN (RAND, TSTMU,  TSTSIG, TSTART)
            CALL NORMGN (RAND, FDERRM, FDERRS, MODER2)

            CALL PRYRV (RAND, EBENDA,EBENDB,LAMPA, LAMPB, EBEND, LAMP)
            CALL PRYRV (RAND, LAMAA, LAMAB, LAMAA, LAMAB, LAMA,  DUM)
            CALL PRYRV (RAND, LAMDAA,LAMDAB,LAMTMA,LAMTMB,LAMDA, LAMTM)
            LAMDA = EXP (LAMDA)
            LAMTM = EXP (LAMTM)

            MINPHI = 1.0E+36
            IF (VARPHI .EQ. 1) THEN
C              WEIBULL INTRINSIC MATERIALS VARIATION
               DO 230 M = 1, NSYM
                  CALL WEIBGN (BZERO, RAND, PHI)
                  MINPHI = MIN (PHI, MINPHI)
  230          CONTINUE
               PHI = MINPHI
            ELSE
C              LOGNORMAL INTRINSIC MATERIALS VARIATION
               DO 231 M = 1, NSYM
                  CALL NORMGN (RAND, 0.0, PHISIG, LNPHI)
                  MINPHI = MIN (LNPHI, MINPHI)
  231          CONTINUE
               PHI = EXP (MINPHI)
            ENDIF

            IF (VARY .EQ. 0) PHI = 1.0

            IF (IOUT .EQ. 15) THEN
               WRITE(8,*) 'HGAS =', HGAS, ' TGAS =', TGAS
               WRITE(8,*) 'SLOPE =', SLOPE, ' LAMG = ', LAMG
               WRITE(8,*) 'LAMP =',LAMP,' EBEND =',EBEND,' LAMA =',LAMA
               WRITE(8,*) 'SPEED = ', SPEED,' LAMDA =',LAMDA
               WRITE(8,*) 'LAMTM =',LAMTM,' PHI =',PHI
               WRITE(8,*) 'MANAL =',MANAL,' TANAL =',TANAL
               WRITE(8,*) 'TSTART =',TSTART,' MODER1 =',MODER1,
     &                    ' MODER2 =',MODER2
            ENDIF

C ** CALCULATE REGION DEPENDENT S/N CURVE PARAMETERS

            FACTR = PHI * KRATIO * Z

            DO 235 L = ZROREG, NUMREG
              TRSBND(L) = FACTR * SBND(L)
              TRBIGK(L) = BIGK(L)
  235       CONTINUE
            TRSBND(0) = SBND(0)

            IF (ZROREG .EQ. 0) CALL KOMO (SZERO, BIGK, MM, NBND,
     &         TRSBND, TRBIGK, FACTR, NUMREG)

            DO 250 L = ZROREG, NUMREG
               LNA(L) = MM(L) * ALOG(TRBIGK(L))
               LPHIM(L) = MM(L) * ALOG(PHI)
               IF (IOUT .EQ. 15) THEN
                  WRITE(8,*) 'L =',L,' MM =',MM(L),' BIGK =',TRBIGK(L)
                  WRITE(8,*) 'LNA =', LNA(L), ' PHI =', PHI
                  WRITE(8,*) 'LPHIM =', LPHIM(L), ' SBND =', SBND(L)
```

```
                 WRITE(8,*) 'KRATIO = ', KRATIO, ' Z = ', Z
                 WRITE(8,*) 'TRSBND =', TRSBND(L), ' FACTR = ', FACTR
              ENDIF
 250        CONTINUE

C ** CALL BLDLIF TO OBTAIN BLADE LCF LIFE

            NEWLIF = LAMDA * LAMTM * BLDLIF (TGAS, HGAS, FAA, FAB, FAC,
     &               FAD, FAE, FAF, MODER1, RPM, TSUBI, SPEED, SLOPE,
     &               TSTART, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
     &               MODER2, FD2A, FD2B, FD3A, FD3B, ETHNOM, MANAL,
     &               LAMP, NOMSPD, EMNOM, TANAL, LAMA, LAMG, EBEND,
     &               NTIME, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM,
     &               KRATIO, LNZ, TRSBND, ZROREG, NUMREG, SZERO)

            IF (IOUT .EQ. 15) WRITE(8,*) 'NEWLIF = ', NEWLIF

            IF (NLIFET .GE. 100) CALL INSORT (NEWLIF, LIFE, NLIFET)

 200     CONTINUE

 150 CONTINUE

      IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES TO FILE LOWLIF

         DO 300 J = 1, (NLIFET / 100)
            WRITE(9,*) J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
 300     CONTINUE

C ** INITIALIZE VARIABLE BLFPOS()

         DO 325 J = 1, MAXBLF
            BLFPOS(J) = 0
 325     CONTINUE

         FIFTY = 0.50E0

C ** PRINT EMPIRICAL BLIVES

         IF (VARPHI .EQ. 1) THEN
            WRITE(3,925)
         ELSE
            WRITE(3,927)
         ENDIF

         DO 350 J = 1, NBLIFE
            BLFPOS(J) = NINT (BLFPER(J) * FLOAT (NLIFET))
            WRITE(3,926) BLFPER(J), LIFE(BLFPOS(J))
 350     CONTINUE
         WRITE(3,926) FIFTY, LIFE(NLIFET/2)

      ENDIF

C ** CALCULATE NORMAL MEDIAN CURVE IF DESIRED

      IF ((VARY .EQ. 3) .AND. (NMED .EQ. 1)) THEN

         CALL SORTM (ALLM, NUMREG, MCOUNT)

         MID = MCOUNT / 2
         DO 400 L = 1, NUMREG
            MEDM(L) = ALLM(MID,L)
 400     CONTINUE

         CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG,
     &                NBND, BIGK1, BZERO)

      ENDIF


C ** FORMAT STATEMENTS TO ECHO INPUT DATA TO BLDLCO
```

```
  900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &        'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &        'NASA Contract NAS7-918 is acknowledged.',////,
     &        33X,'INPUT DATA',
     &        ///,14X,'DRIVERS',25X,'PARAMETER DISTRIBUTIONS',
     &        //,48X,'RHO',16X,'THETA')

  901 FORMAT(/,2X,'Hgas',13X,'Be(',F5.0,',',F6.0,')',5X,
     &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')',
     &        //,2X,'Tgas (deg R)',5X,'Be(',F5.0,',',F6.0,')',5X,
     &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')

  902 FORMAT(/,2X,'DECEL SLOPE',6X,'Be(',F5.0,',',F6.0,')',5X,
     &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')',
     &        //,2X,'Tgas UNCERT.',5X,'Be(',F5.2,',',F6.2,')',5X,
     &        'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')

  903 FORMAT(//,50X,'N( MEAN, STD. DEV.)',
     &        //,2X,'ROTOR SPEED VARIATION (rpm) AT TIME T',I1,
     &        10X,'N(',F8.1 ,',',F7.1,')',//,
     &        2X,'Faccel MODELING ERROR',27X,'N(',F4.1,',',E11.4,')',
     &        //,2X,'STARTING DECEL TEMPERATURE (deg R)',14X,
     &        'N(',F8.2,',',F7.2,')',//,
     &        2X,'Fdecel MODELING ERROR',27X,'N(',F4.1,',',E11.4,')')

  904 FORMAT(//,2X,'STRAIN DUE TO GAS BENDING (%)',17X,
     &        'U(',F8.5,',',F9.5,')',
     &        //,2X,'LAMBDA BLADE PULL',29X,
     &        'U(',F8.5,',',F9.5,')',
     &        //,2X,'MECHANICAL ANALYSIS FACTOR',20X,
     &        'U(',F8.5,',',F9.5,')',
     &        //,2X,'COEFFICIENT OF THERMAL EXPANSION FACTOR',7X,
     &        'U(',F8.5,',',F9.5,')',
     &        //,2X,'THERMAL ANALYSIS FACTOR',23X,
     &        'U(',F8.5,',',F9.5,')')

  905 FORMAT(//,2X,'DAMAGE MODEL ACCURACY',21X,
     &        'U(ln',F8.5,', ln',F8.5,')',
     &        //,2X,'TMF MODEL ACCURACY',24X,
     &        'U(ln',F8.5,', ln',F8.5,')')

  906 FORMAT(////,20X,'OTHER STRAIN HISTORY INPUT',
     &        ///,2X,'NOMINAL MECHANICAL STRAIN (%)',23X,F6.4,
     &        //,2X,'NOMINAL ROTOR SPEED (rpm)',23X,F6.0,
     &        //,2X,'STRAIN-TIME HISTORY PERIOD (missions)',14X,F5.2,
     &        //,2X,'STRAIN-TIME HISTORY NOISE FILTER (%)',16X,F7.5,
     &        //,2X,'NUMBER OF POINTS IN HISTORIES',19X,I5,
     &        //,2X,'WALKER EXPONENT',36X,F5.2)

  907 FORMAT(////,6X,'COEFFICIENTS OF ACCELERATION AND DECELERATION ',
     &        'FUNCTIONS',//,2X,'THERMAL STRAIN AT STARTUP (%):',/,5X,
     &        'Faccel(Tgas, Hgas) = ',E13.6,' + ',E13.6,' * Tgas + ',
     &        /,15X,E13.6,' * Hgas + ',E13.6,' * Tgas ** 2 + ',
     &        /,15X,E13.6,' * Hgas**2 + ',E13.6,' * Tgas * Hgas',
     &        //,2X,'THERMAL STRAIN AT SHUTDOWN (%):',/,5X,
     &        'Fdecel1(m, Tstart) = ',E13.6,' + ',E13.6,' * Tstart + ',
     &        /,15X,E13.6,' * m + ',E13.6,' * Tstart ** 2 + ',
     &        /,15X,E13.6,' * m ** 2 + ',E13.6,' * Tstart * m',
     &        //,2X,'TIME AT SHUTDOWN (sec):',
     &        /,5X,'Fdecel2(m, Tstart) = ',E13.6,' + ','(Tstart - ',
     &        E13.6,' ) / m',
     &        //,2X,'ROTOR SPEED AT SHUTDOWN (rpm):',
     &        /,5X,'Fdecel3(t) = ',E13.6,' + ',E13.6,' * t',
     &        /////,20X,'STRAIN HISTORY INFORMATION',
     &        ///,5X,'ROTOR SPEED',5X,'THERMAL STRAIN',
     &        /,9X,'rpm',15X,'(%)',/)

  908 FORMAT(7X,F7.1,9X,F9.6)

  925 FORMAT(///,2X,'   WEIBULL VARIATION',
     &        //,2X,'B LIVES:        EMPIRICAL',/)

  926 FORMAT(2X,F7.5,5X,E13.6)

  927 FORMAT(///,2X,'  LOGNORMAL VARIATION',
```

```
&        //,2X,'B LIVES:      EMPIRICAL',/)

      STOP
      END

C***********************************************************************
C                   SAMPLE 'BLDLCD' INPUT FILE
C***********************************************************************
C 675...............................RANDOM NUMBER SEED
C 0.................................OUTPUT DUMP CONTROLLER
C 100...............................INNER LOOP SIZE
C 200...............................OUTER LOOP SIZE
C 50................................SYMMETRY NUMBER
C 2.................................UNIFORM S/N VARIATION
C 0.................................NORMAL MEDIAN NOT REQUIRED
C 0.................................MAT. PROC. VAR. NOT REQUIRED
C 1.................................WEIBULL INTRINSIC VARIATION
C 3.................................NUMBER OF BLIVES REQUESTED
C 0.0001............................B.01 LIFE
C 0.001.............................B.1 LIFE
C 0.01..............................B1 LIFE
C  676. 2730. 0.5  0.5  0.0  0.0....Hgas           (A,B) (R1,R2) (T1,T2)
C  800. 2000. 0.5  0.5  0.0  0.0....Tgas           (A,B) (R1,R2) (T1,T2)
C 2730. 2730. 0.5  0.5  0.0  0.0....DECEL SLOPE    (A,B) (R1,R2) (T1,T2)
C 0.80  1.20  0.5  0.5  0.0  0.0....Tgas UNCERTAINTY FACTOR
C 5   37592.   507..................ROTOR SPEED VARIATION PARAMETERS:
C                                     i, MEAN, STD.DEV. (NORMAL DIST.)
C 0.0     0.020.....................Faccel MODELING ERROR MEAN & STD.DEV.
C 1640.0   40.67....................DECEL Tstart MEAN & STANDARD DEVIATION
C 975.3    28.6.....................STANDARD RESPONSE PROBE MEAN & STD DEV
C 0.0    0.003......................Fdecel MODELING ERROR MEAN & STD DEV
C 0.0      0.0......................STRAIN DUE TO GAS BENDING (%)
C 0.96     1.04.....................LAMBDA BLADE PULL
C 0.80     1.20.....................MECHANICAL ANALYSIS ACCURACY FACTOR
C 0.975    1.025....................COEFFICIENT OF THERMAL EXPANSION
C 0.70     1.30.....................THERMAL ANALYSIS ACCURACY FACTOR
C -0.693147   0.563283..............DAMAGE ACCUMULATION MODEL ACCURACY
C  0.00       0.00..................TMF MODEL ACCURACY
C 0.295   38482.....................NOMINAL MECH. STRAIN & ROTOR SPEED (%,RPM)
C 1.0...............................STRAIN-TIME HISTORY PERIOD (MISSIONS)
C 0.000.............................STRAIN-TIME HISTORY NOISE FILTER (%)
C 6.................................NUMBER OF POINTS IN STRAIN-TIME HISTORY
C 0.5...............................WALKER EXPONENT
C
C COEFFICIENTS FOR STARTUP RESPONSE SURFACE FOR THERMAL STRAIN:
C    Faccel(Tgas,Hgas) = A + B * T + C * H + D * T**2 + E * H**2 + F * T * H
C     A          B            C            D            E            F
C 0.00727362 0.000067442 -0.000059109 -3.52929E-08 1.07611E-08 -2.74419E-08
C
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR THERMAL STRAIN:
C    Fdecel1(m,Tstart) = A + B * Tstart + C * m + D * Tstart ** 2
C                          + E * m ** 2 + F * Tstart * m
C     A             B              C            D       E          F
C -0.132623     0.000227427    -0.000059290    0.00    0.00    4.71714E-08
C
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR TIME:
C    Fdecel2(m,Tstart) = A + (Tstart - B) / m
C     A            B
C   0.20         950.0
C
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR RPM:
c    Fdecel3(t) = A + B * t
C     A              B
C 30523.07      -21846.15
C
C RPM(TIME)   THERMAL STRAIN (%).....STRAIN HISTORY INFORMATION
C   225.8          0.0
C  3025.1         -0.196921
C  6138.8          0.146025
C  8309.0         -0.200128
C     0.0          0.007393
```

```
C
C 'RT, PWA 1480, 001 DIRECTION'.....SPECIFIC MATERIAL DESCRIPTION
C 1.54    1.57    1    8...............YIELD & ULTIMATE STRENGTHS, NDIV, NPTS
C 8      -1.0     1...................# PTS IN DIV, STRAIN RATIO, REGION
C 0.89       6800.....................S(1) N(1)        RAW
C 0.89      15000.....................S(2) N(2)    STRAIN-LIFE
C 0.67      27000.....................S(3) N(3)       (S/N)
C 0.67      43200.....................S(4) N(4)        DATA
C 0.56     139300.....................S(5) N(5)       POINTS
C 0.56     545200.....................S(6) N(6)      FOR THE
C 0.56     147000.....................S(7) N(7)      SPECIFIC
C 0.39    4344800.....................S(8) N(8)      MATERIAL
C 0.00................................NO VALUE OF So SUPPLIED (%)
C 1  0................................NUMBER OF REGIONS:W/DATA W/O DATA
C 1.0E+36.............................LIFE BOUNDARIES:  REGION 1
C 0.00................................
C 0    0.00      0.00.................CONSTRAINT ON COEFF. OF VARIATION
C 0.0     0.0    0.0..................0 PTS IN RANGE, LOWER BOUND, UPPER BOUND
C                                     NORMAL DIST. PRIORS: DELTA, Mo, SIGMA2
C
C**********************************************************************
C                           LIST OF VARIABLES
C**********************************************************************
C
C ALLM()      2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C BIGK()      1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C BIGK1       EQUAL TO BIGK(1) - DUMMY PARAMETER FOR CALLS TO SUBROUTINE
C               EXPCTD
C BLDLIF      REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND LCF
C               LIFE CALCULATION
C BLFPER()    1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C BLFPOS()    1-D ARRAY CONTAINING POSITION IN LIFE() OF EMPIRICAL BLIVES
C BZERO       VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING S/N DATA SET
C DUM         DUMMY VARIABLE
C EBEND       SELECTED VALUE FOR BENDING STRAIN (%)
C EBENDA      EBEND LOWER BOUND
C EBENDB      EBEND UPPER BOUND
C EMNOM       NOMINAL MECHANICAL STRAIN (%)
C ETHNOM()    1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY
C FAA, FAB, FAC, FAD, FAE, FAF
C             COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
C FACTR       SCALE FACTOR EQUAL TO PHI * KRATIO * Z
C FAERRM      STARTUP THERMAL STRAIN RESPONSE SURFACE MEAN
C FAERRS      STARTUP THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
C FD1A, FD1B, FD1C, FD1D, FD1E, FD1F
C             COEFFICIENTS FOR FD1, ONE OF THE DECELERATION FUNCTIONS
C FD2A, FD2B
C             COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS
C FD3A, FD3B
C             COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS
C FDERRM      DECELERATION THERMAL STRAIN RESPONSE SURFACE MEAN
C FDERRS      DECELERATION THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
C FIFTY       EQUAL TO .5 - USED TO ACCESS 50% POINT IN LIFE()
C FTU         MATERIAL ULTIMATE STRENGTH (%)
C FTY         MATERIAL YIELD STRENGTH (%)
C HGAS        SELECTED HOT GAS FILM COEFFICIENT, Hgas
C HGASA       HGAS LOWER BOUND
C HGASB       HGAS UPPER BOUND
C HGASR       SELECTED RHO FOR HGAS
C HGASR1      HGAS - RHO LOWER BOUND
C HGASR2      HGAS - RHO UPPER BOUND
C HGAST       SELECTED THETA FOR HGAS
C HGAST1      HGAS - THETA LOWER BOUND
C HGAST2      HGAS - THETA UPPER BOUND
C I           CONTROLS INNER DO LOOP
C IOUT        CONTROLS DUMP TO FILE IOUTPR
C J           CONTROLS DO LOOP FOR EACH BLIFE
C K           CONTROLS OUTER DO LOOP
C KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C L           CONTROLS DO LOOP FOR EACH REGION
C LAMA        SELECTED COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR,
C               LAMbda Alpha
C LAMAA       LAMA LOWER BOUND
C LAMAB       LAMA UPPER BOUND
C LAMDA       SELECTED DAMAGE ACCUMULATION MODEL ACCURACY FACTOR, LAMbda
```

```
C                Damage Accumulation
C LAMDAA       LAMDA LOWER BOUND
C LAMDAB       LAMDA UPPER BOUND
C LAMG         SELECTED UNCERTAINTY IN Tgas
C LAMGA        LAMG LOWER BOUND
C LAMGB        LAMG UPPER BOUND
C LAMGR        SELECTED RHO FOR LAMG
C LAMGR1       LAMG - RHO LOWER BOUND
C LAMGR2       LAMG - RHO UPPER BOUND
C LAMGT        SELECTED THETA FOR LAMG
C LAMGT1       LAMG - THETA LOWER BOUND
C LAMGT2       LAMG - THETA UPPER BOUND
C LAMP         SELECTED DEVIATION IN BLADE PULL DUE TO BLADE MASS, LAMbda
C                Pull
C LAMPA        LAMP LOWER BOUND
C LAMPB        LAMP UPPER BOUND
C LAMTM        SELECTED TMF MODEL ACCURACY FACTOR, LAMbda TMf
C LAMTMA       LAMTM LOWER BOUND
C LAMTMB       LAMTM UPPER BOUND
C LIFE()       1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
C                - SORTED VALUES OF THE LEFT-HAND TAIL
C LNA()        1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
C LNPHI        LOGNORMAL(0,PHISIG**2) GENERATED RANDOM VARIATE
C LNZ          NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
C LPHIM()      1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
C M            CONTROLS SYMMETRY DO LOOP
C MANAL        SELECTED MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
C MANALA       MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR LOWER BOUND
C MANALB       MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR UPPER BOUND
C MAXBLF       MAXIMUM NUMBER OF BLIVES TO BE PROVIDED
C MAXDAT       MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED
C MAXLIF       MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA,
C                ALPHA CALCULATION
C MAXM         MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY
C MAXMM        MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION
C MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C MCOUNT       NUMBER OF M's TO BE USED TO CALCULATE THE TRUNCATED NORMAL
C                MEDIAN S/N CURVE
C MEDM()       1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C MID          POINTER TO THE MEDIAN M VALUES - EQUAL TO HALF OF MCOUNT
C MINPHI       EQUAL TO MIN(PHI) - THE MINIMUM OF NSYM DRAWS OF PHI
C MM()         1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C MODER1       MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
C MODER2       MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE
C MPROC        Materials PROCess variation - CONTROLS MATERIALS PROCESS
C                VARIATION - 0 - NO VARIATION;  1 - VARIATION
C MU()         1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C NBLIFE       NUMBER OF BLIVES TO BE PROVIDED
C NBND()       1-D ARRAY CONTAINING UPPER BOUNDS FOR THE NUMREG LIFE
C                REGIONS OF INTEREST FOR THE SPECIFIC (REFERENCE) MATERIAL
C                S/N DATA SET
C NEWLIF       LIFE VALUE RETURNED FROM CALL TO BLDLIF
C NF()         2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL
C                S/N DATA SET BROKEN INTO LIFE REGIONS
C NHYPER       SIZE OF OUTER LOOP
C NLIFE        SIZE OF INNER LOOP
C NLIFET       TOTAL NUMBER OF LIVES CALCULATED BY PFM
C NMED         CONTROLS MEDIAN CALCULATION FOR THE TRUNCATED NORMAL
C                DISTRIBUTION CASE - 0 - NO MEDIAN CALCULATION;
C                1 - MEDIAN CALCULATION DESIRED
C NOMSPD       NOMINAL ROTOR SPEED, RPM
C NPTS()       1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION
C                FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
C NSYM         SYMMETRY NUMBER
C NTIME        NUMBER OF POINTS IN STRAIN-TIME HISTORY
C NUMREG       NUMBER OF REGIONS OF INTEREST
C PERIOD       LENGTH OF TIME IN MISSIONS OF TIME HISTORY
C PHI          WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C PHISIG       EQUAL TO PI * (6 ** .5) / BZERO - VALUE OF LOGNORMAL
C                PARAMETER, SIGMA, CHARACTERIZING S/N DATA SET
C PSIG         EQUAL TO SQRT(PVAR) - MATERIALS PROCESS STANDARD DEVIATION
C PVAR         MATERIALS PROCESS VARIATION
C RAND         RANDOM NUMBER SEED
C RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON M FOR
```

```
C                   EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L)
C                   IS THE UPPER BOUND
C RPM()           1-D ARRAY CONTAINING ROTOR SPEED HISTORY (rpm)
C SBND()          1-D ARRAY CONTAINING THE STRAIN VALUES (%, R = -1.0)
C                   CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C                   REGION CONTAINED IN NBND()
C SIG()           1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                   DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C SLOPE           SELECTED DECELERATION SLOPE, m (deg R / sec)
C SLOPEA          m LOWER BOUND
C SLOPEB          m UPPER BOUND
C SLOPR           SELECTED RHO FOR m
C SLOPR1          m - RHO LOWER BOUND
C SLOPR2          m - RHO UPPER BOUND
C SLOPT           SELECTED THETA FOR m
C SLOPT1          m - THETA LOWER BOUND
C SLOPT2          m - THETA UPPER BOUND
C SPEED           SELECTED STEADY STATE ROTOR SPEED, RPM
C SPEEDM          MEAN OF ROTOR SPEED (MU, NORMAL DISTRIBUTION)
C SPEEDS          STANDARD DEVIATION OF ROTOR SPEED (SIGMA, NORMAL DISTRIBUTION)
C STR()           2-D ARRAY CONTAINING STRAIN POINTS (STRAIN RATIO = -1.0) FOR
C                   THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS
C SZERO           STRAIN TENSILE TEST POINT, So
C TANAL           SELECTED THERMAL STRAIN ANALYSIS ACCURACY FACTOR
C TANALA          THERMAL STRAIN ANALYSIS ACCURACY FACTOR LOWER BOUND
C TANALB          THERMAL STRAIN ANALYSIS ACCURACY FACTOR UPPER BOUND
C TGAS            SELECTED GAS TEMPERATURE Tgas
C TGASA           GAS TEMPERATURE LOWER BOUND
C TGASB           GAS TEMPERATURE UPPER BOUND
C TGASR           SELECTED RHO FOR GAS TEMPERATURE
C TGASR1          GAS TEMPERATURE - RHO LOWER BOUND
C TGASR2          GAS TEMPERATURE - RHO UPPER BOUND
C TGAST           SELECTED THETA FOR GAS TEMPERATURE
C TGAST1          GAS TEMPERATURE - THETA LOWER BOUND
C TGAST2          GAS TEMPERATURE - THETA UPPER BOUND
C TRBIGK()        1-D ARRAY CONTAINING VALUES OF BIGK() CORRECTED FOR SZERO,
C                   PHI, KRATIO, AND Z
C TRSBND()        1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR
C                   EACH REGION CALCULATED FOR EACH TRIAL
C TRUNC           VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)
C TSTART          STARTING DECELERATION TEMPERATURE (deg R)
C TSTMU           MEAN OF TSTART
C TSTSIG          STANDARD DEVIATION OF TSTART
C TSUBI           THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS
C VARPHI          CONTROLS TYPE OF INTRINSIC MATERIALS VARIATION DESIRED -
C                   1 - WEIBULL VARIATION;  2 - LOGNORMAL VARIATION
C VARY            CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO VARIATION;
C                   1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;  3 - TRUN-
C                   CATED NORMAL VARIATION
C WEXP            WALKER EXPONENT
C Z               LOGNORMAL(0,PVAR) GENERATED RANDOM VARIATE
C ZROREG          ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                   BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION


C***********************************************************************


C  FUNCTION BLDLIF PERFORMS THE DRIVER TRANSFORMATION AND CALLS RAINF3
C  TO CALCULATE THE FATIGUE LIFE
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JAN92      COMMENTS:  3APR92
C     VERSION:  BLDLCF 3.4 (MATCHR V8.5, RAINF3 V1.1)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      FUNCTION BLDLIF (TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF,
     &                 MODER1, RPM, TSUBI, SPEED, SLOPE, TSTART, FD1A,
     &                 FD1B, FD1C, FD1D, FD1E, FD1F, MODER2, FD2A,
     &                 FD2B, FD3A, FD3B, ETHNOM, MANAL, LAMP, NOMSPD,
     &                 EMNOM, TANAL, LAMA, LAMG, EBEND, NTIME, TRUNC,
```

```
     &                PERIOD, WEXP, MM, LNA, LPHIM, KRATIO, LNZ,
     &                TRSBND, ZROREG, NUMREG, SZERO)

C  SUBPROGRAMS:  RAINF3
C       INPUTS:  TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF, MODER1, RPM,
C                TSUBI, SPEED, SLOPE, TSTART, FD1A, FD1B, FD1C, FD1D,
C                FD1E, FD1F, MODER2, FD2A, FD2B, FD3A, FD3B, ETHNOM, MANAL,
C                LAMP, NOMSPD, EMNOM, TANAL, LAMA, LAMG, EBEND, NTIME,
C                TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO, LNZ, TRSBND,
C                ZROREG, NUMREG, SZERO
C      OUTPUTS:  BLDLIF

C     IMPLICIT NONE

      INTEGER  MAXM, MAXREG

      PARAMETER (MAXM = 50, MAXREG = 3)

      COMMON   IOUT

      INTEGER  I, IOUT, NTIME, NUMREG, TSUBI, ZROREG

      REAL     BLDLIF, EBEND, EM(MAXM), EMNOM, ETH(MAXM), ETHNOM(MAXM),
     &         ETOT(MAXM), FA, FAA, FAB, FAC, FAD, FAE, FAF, FD1,
     &         FD1A, FD1B, FD1C, FD1D, FD1E, FD1F, FD2, FD2A, FD2B,
     &         FD3, FD3A, FD3B, HGAS, KRATIO, LAMA, LAMG, LAMP,
     &         LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MANAL,
     &         MM(0:MAXREG), MODER1, MODER2, NOMSPD, PERIOD, RAINF3,
     &         RPM(MAXM), SLOPE, SPEED, SZERO, TANAL, TGAS,
     &         TRSBND(0:MAXREG), TRUNC, TSTART, WEXP

C                          LIST OF VARIABLES
C
C EBEND        SELECTED VALUE FOR BENDING STRAIN (%)
C EM()         1-D ARRAY CONTAINING THE SIMULATED MECHANICAL STRAIN-TIME
C                HISTORY (%)
C EMNOM        NOMINAL MECHANICAL STRAIN (%)
C ETH()        1-D ARRAY CONTAINING THE SIMULATED THERMAL STRAIN-TIME HISTORY
C ETHNOM()     1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY
C ETOT()       1-D ARRAY CONTAINING THE TOTAL STRAIN-TIME HISTOY
C FA           VALUE OF ACCELERATION FUNCTION FOR THERMAL STRAIN - SECOND
C                ORDER POLYNOMIAL AS A FUNCTION OF TGAS AND HGAS
C FAA, FAB, FAC, FAD, FAE, FAF
C              COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
C FD1          VALUE OF DECELERATION FUNCTION FOR THERMAL STRAIN - SECOND
C                ORDER POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE
C FD1A, FD1B, FD1C, FD1D, FD1E, FD1F
C              COEFFICIENTS FOR FD1, ONE OF THE DECELERATION FUNCTIONS
C FD2          VALUE OF DECELERATION FUNCTION FOR TIME - SECOND ORDER
C                POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE
C FD2A, FD2B
C              COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS
C FD3          VALUE OF DECELERATION FUNCTION FOR ROTOR SPEED - FIRST ORDER
C                POLYNOMIAL (LINEAR) FUNCTION OF TIME
C FD3A, FD3B
C              COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS
C HGAS         SELECTED HOT GAS FILM COEFFICIENT, Hgas
C I            CONTROLS DO LOOP FOR EACH POINT IN TIME HISTORY
C IOUT         CONTROLS DUMP TO FILE IOUTPR
C KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C LAMA         SELECTED VALUE FOR COEFFICIENT OF THERMAL EXPANSION ACCURACY
C                FACTOR, LAMbda Alpha
C LAMG         THE UNCERTAINTY IN Tgas
C LAMP         SELECTED VALUE FOR DEVIATION IN BLADE PULL DUE TO BLADE MASS,
C                LAMbda Pull
C LNA()        1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
C LNZ          NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
C LPHIM()      1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
C MANAL        SELECTED VALUE FOR MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
C MAXM         MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY
C MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C MM()         1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C MODER1       MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
```

```
C MODER2       MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE
C NOMSPD       NOMINAL ROTOR SPEED, RPM
C NTIME        NUMBER OF POINTS IN STRAIN-TIME HISTORY
C NUMREG       NUMBER OF REGIONS OF INTEREST
C PERIOD       LENGTH OF TIME IN MISSIONS OF TIME HISTORY
C RAINF3       REAL FUNCTION PERFORMING RAINFlow COUNTING, DAMAGE ACCUMU-
C                LATION AND FATIGUE LIFE PREDICTION (USING THE MATERIALS
C                CHARACTERIZATION MODEL)
C RPM()        1-D ARRAY CONTAINING ROTOR SPEED HISTORY
C SLOPE        SELECTED VALUE FOR DECELERATION SLOPE, deg R / sec
C SPEED        SELECTED VALUE FOR STEADY STATE ROTOR SPEED, rpm
C SZERO        STRAIN TENSILE TEST POINT, So
C TANAL        SELECTED VALUE FOR THERMAL STRAIN ANALYSIS ACCURACY FACTOR
C TGAS         SELECTED VALUE FOR HOT GAS TEMPERATURE Tgas (deg R)
C TRSBND()     1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR
C                EACH REGION CALCULATED FOR EACH TRIAL
C TRUNC        VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)
C TSTART       STARTING DECELERATION TEMPERATURE (deg R)
C TSUBI        THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS
C WEXP         WALKER EXPONENT
C ZROREG       ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                REGION


C ** CALCULATE STRAIN HISTORY

      FA = FAA + FAB * TGAS + FAC * HGAS + FAD * TGAS ** 2
     &     + FAE * HGAS ** 2 + FAF * TGAS * HGAS + MODER1
      ETHNOM(1) = FA

      RPM(TSUBI) = SPEED

      FD1 = FD1A + FD1B * TSTART + FD1C * SLOPE + FD1D * TSTART ** 2
     &      + FD1E * SLOPE ** 2 + FD1F * TSTART * SLOPE + MODER2
      FD2 = FD2A + (TSTART - FD2B) / SLOPE
      FD3 = FD3A + FD3B * FD2
      RPM(NTIME) = FD3
      ETHNOM(NTIME) = FD1

      DO 100 I = 1, NTIME
         EM(I) = MANAL * LAMP * (RPM(I) / NOMSPD) ** 2 * EMNOM
         ETH(I) = TANAL * LAMA * ETHNOM(I)
         IF ((I .GT. 1) .AND. (I .LT. TSUBI))
     &      ETH(I) = LAMG * ETH(I)
         ETOT(I) = EBEND + EM(I) + ETH(I)
  100 CONTINUE

      IF (IOUT .EQ. 15) THEN
         WRITE(8,*) 'FA = ', FA, ' ETHNOM1 = ', ETHNOM(1)
         WRITE(8,*) 'RPMI = ', RPM(TSUBI), ' LAMG = ', LAMG
         WRITE(8,*) 'FD1 = ', FD1, ' FD2 = ', FD2
         WRITE(8,*) 'FD3 = ', FD3, ' RPM = ', RPM(NTIME)
         WRITE(8,*) '    ETHNOM = ', ETHNOM(NTIME)
         DO 125 I = 1, NTIME
            WRITE(8,*) 'I = ', I, ' EM = ', EM(I)
            WRITE(8,*) 'ETH = ', ETH(I), ' ETOT = ', ETOT(I)
  125    CONTINUE
      ENDIF

C ** CALL RAINF3 TO CALCULATE DAMAGE AND RESULTING FATIGUE LIFE

      BLDLIF = RAINF3 (ETOT, NTIME, TRUNC, PERIOD, WEXP, MM, LNA,
     &                 LPHIM, KRATIO, LNZ, TRSBND, ZROREG, NUMREG,
     &                 SZERO)


      RETURN
      END


C***********************************************************************
```

```
C  SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20JUL90
C     VERSION:  2.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

C  INPUTS:   NEWLIF, LIFE, NLIFET
C  OUTPUTS:  LIFE

C     IMPLICIT NONE

      INTEGER  MAXLIF

      PARAMETER (MAXLIF = 10000)

      COMMON   IOUT

      INTEGER  I, IOUT, NLIFET, NUM, PLACE

      REAL     LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)


C                   LIST OF VARIABLES
C
C  I        CONTROLS DO LOOP FOR INSERTION
C  IOUT     OUTPUT DUMP CONTROLLER
C  LIFE()   1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
C             PFM TO BE SORTED
C  MAXLIF   MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
C             CALCULATION
C  NEWLIF   LIFE VALUE TO BE INSERTED INTO LIFE()
C  NLIFET   TOTAL NUMBER OF LIVES CALCULATED BY PFM
C  NUM      NUMBER OF LIFE VALUES IN LIFE()
C  PLACE    POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
C  TEMP()   1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON
C             INSERTION OF NEWLIF


      NUM = NLIFET / 2

C     FIND POSITION IN LIFE() FOR NEWLIF

      IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

      DO 100 I = 1, NUM
         IF (NEWLIF .LT. LIFE(I)) THEN
            PLACE = I
            GOTO 110
         ENDIF
  100 CONTINUE
  110 CONTINUE

C     STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()

      DO 200 I = (PLACE + 1), NUM
         TEMP(I) = LIFE(I-1)
  200 CONTINUE

C     INSERT NEWLIF

      LIFE(PLACE) = NEWLIF

C     SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

      DO 300 I = (PLACE + 1), NUM
         LIFE(I) = TEMP(I)
  300 CONTINUE
```

```
C     IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN
  400 CONTINUE

      RETURN
      END

C***********************************************************************
C   SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C   INDEPENDENT RANDOM VARIATES
C   PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C         DATE:  9MAR87
C   SUBPROGRAM:  RANDOM
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, RHO1, RHO2, THE1, THE2, X, Y

      INTEGER IOUT

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      X = FRAC * (RHO2 - RHO1) + RHO1

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      Y = FRAC * (THE2 - THE1) + THE1

      IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ' RHO2 =', RHO2,
     &        ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y

      RETURN
      END


C***********************************************************************
C  THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  9MAR87
C  SUBPROGRAM:  GAM
C
C  The random variates are generated using the method described in:
C  Johnson, N. L., and Kotz, S., Distribution in Statistics: Continuous
C    Univariate Distributions - 1, Houghton Mifflin Company, 1970,
C    pp. 181-182.
C***********************************************************************

      SUBROUTINE BETAGN (RAND, RHO, THETA, A, B, X)

      COMMON IOUT

      DOUBLE PRECISION RAND

      REAL    A, B, GAM, RHO, THETA, W, X, Y1, Y2

      INTEGER IOUT

      IF (IOUT .EQ. 15) WRITE(8,*) 'RAND =', RAND, ' RHO =', RHO,
     &        ' THETA =', THETA, ' A =', A, ' B =', B, ' X =', X
      Y1 = GAM((RHO * THETA + 1.), RAND)
      Y2 = GAM(((1. - RHO) * THETA + 1.), RAND)
```

```
      W = Y1 / (Y1 + Y2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'Y1 =', Y1, ' Y2 =', Y2, ' W =', W

C  TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

      X = W * (B - A) + A
      IF (IOUT .EQ. 15) WRITE(8,*) 'W =', W, ' X =', X

      RETURN
      END

C***********************************************************************

C  The random variates are generated using an "Acceptance/Rejection Method"
C  Fishman, George S., "Sampling From the Gamma Distribution on a
C    Computer," Communications of the ACM, Volume 19, Number 7, July 1976,
C    pp. 407-409.

      REAL FUNCTION GAM (ALPHA, RAND)

C     SUBPROGRAM:  RANDOM

      COMMON  IOUT

      INTEGER IOUT

      REAL    A, ALPHA, ARG, U1, U2, V1, V2

      DOUBLE PRECISION RAND

      A = ALPHA - 1.
C     IF (IOUT .EQ. 15) WRITE(8,*) 'A =', A, ' ALPHA =', ALPHA

   10 CALL RANDOM (U1, RAND)
      CALL RANDOM (U2, RAND)
      V1 = - ALOG(U1)
      V2 = - ALOG(U2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =',
C    &                   V1, ' V2 =', V2
      ARG = A * (V1 - ALOG(V1) - 1.)
      IF (V2 .LT. ARG) GOTO 10

      GAM = ALPHA * V1
C     IF (IOUT .EQ. 15) WRITE(8,*) 'GAMMA =', GAM

      RETURN
      END


C***********************************************************************

C  SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
C  AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C  FOR THE STRESS FORMULATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  13JUL89      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V8.4, V8.5    MATGRM V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG,
     &                   NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC,
     &                   KRATIO, PVAR)

C  INPUTS:    READS DATA FROM SPECFD AND RELATD; VARY, MPROC
C  OUTPUTS:   RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG, NUMREG,
C             NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
C  SUBPROGRAMS:  INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDREG,
C                CONCAV, MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR
```

```
C        FILES:  5:RELATD-OLD; 6:RELATO-NEW

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
     &        NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
     &        NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG

      REAL    BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
     &        FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
     &        KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
     &        MCHAT(2, MAXREG), MEDM(MAXREG), MO(MAXREG), MU(MAXREG),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        PVAR, RANGEM(2, MAXREG), RATSTR(MAXDAT, 0:MAXSET),
     &        RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
     &        SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT, MAXREG),
     &        SUHAT2(MAXREG), SWHAT2(MAXREG), SX2(MAXREG),
     &        SXY(MAXREG), SY2(MAXREG), SZERO


C                        LIST OF VARIABLES
C
C  BIGKHT      EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
C  BZERO       VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE S/N
C                DATA SET
C  CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
C                AND SIG() CALCULATION
C  FTUZ        ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  IOUT        OUTPUT DUMP CONTROLLER
C  IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                FOR EACH REGION
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LAMN        LAMBDA-N - RATIO OF Var(Ln N given S) / (m**2 C**2),
C                CONSTANT OVER REGIONS AND COMPONENTS
C  LNNF()      3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()     3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                BOUND
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MC() FOR EACH REGION
C  MEDM()      1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C  MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C  MPROC       Materials PROCess variation -CONTROLS MATERIALS PROCESS
C                VARIATION - 0 - NO VARIATION;  1 - VARIATION
C  MU()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
```

```
C                 IS THE UPPER BOUND
C   NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST
C   NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                 SET IN EACH REGION
C   NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                 ALL DATA SETS IN A REGION (Number of Points Per Region)
C   NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C   NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   PVAR        MATERIALS PROCESS VARIATION
C   RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C   RATSTR()    2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
C                 STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C   RAWNF()     2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                 DATA SETS
C   RAWSTR()    2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
C                 DATA (%) FOR ALL S/N DATA SETS
C   REFNP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C   SIG()       1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C   SIGMA2()    1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 VARIANCE FOR EACH REGION
C   STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SUHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SWHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SX2()       1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                 (X = Ln S)
C   SXY()       1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR EACH
C                 REGION (X = Ln S, Y = Ln N)
C   SY2()       1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                 (Y = Ln N)
C   SZERO       STRESS TENSILE TEST POINT, So
C   VARY        CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO
C                 VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                 VARIATION;  3 - TRUNCATED NORMAL VARIATION
C   ZROREG      ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C               BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

      OPEN(5, FILE = 'RELATD', STATUS = 'OLD')
      OPEN(6, FILE = 'RELATO', STATUS = 'NEW')

C  RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C  RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET
C  INFORMATION

C  PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION

C  INITIALIZE PRIMARY ARRAYS

      CALL INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
     &           NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

C  READ, CONVERT, ECHO INFORMATION

      CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR,
     &          LNNF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, NNODAT,
     &          NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO,
     &          SIGMA2, KRATIO, LAMN)

C  CALCULATE RESIDUAL VARIANCES

      CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD,
     &             SWHAT2, SUHAT2, NPPR)
```

```
C  CALCULATE M CONTRAINT BASED ON Co

      CALL FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

      IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1) .OR. (VARY .EQ. 2)) THEN

C  CALCULATIONS FOR ALL TYPES OF VARIATION SAVE NORMAL

C  CALCULATE BOUNDS FOR CONFIDENCE INTERVALS

         CALL INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                JZERO, MCHAT)

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C  OBTAIN POSTERIOR RANGES ON M

         CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT,
     &                RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

         CALL ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

         CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

         WRITE(7,900)

         DO 25 L = 1, NUMREG
            WRITE(7,905) L, IZERO(1, L), IZERO(2, L),
     &                      JZERO(1, L), JZERO(2, L)
   25    CONTINUE

         WRITE(7,910)

         DO 50 L = 1, NUMREG
            WRITE(7,915) L, MCHAT(2,L), MCHAT(1,L)
   50    CONTINUE

         IF (CZERO .GT. 0.0) THEN
            WRITE(7,960)
            DO 150 L = 1, NUMREG
               IF (MCPNT(L) .EQ. 1) THEN
                  WRITE(7,965) L, MC(1,L)
               ELSEIF (MCPNT(L) .EQ. 2) THEN
                  WRITE(7,970) L, MC(1,L), MC(2,L)
               ENDIF
  150       CONTINUE
         ENDIF

         WRITE(7,920)
         WRITE(7,930)

         DO 100 L = 1, NUMREG
            WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
  100    CONTINUE

         WRITE(7,950)

C  CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO

         CALL MEDIAN (NUMREG, RANGEM, MEDM)

C  CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS
```

```
      CALL EXPCTD (1, MEDM, REFNP, STR, NF, SZERO, NUMREG, ZROREG,
     &                NBND, BIGKHT, BZERO)

C  CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED

      IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
         DO 200 L = 1, NUMREG
            RANGEM(1,L) = MEDM(L)
            RANGEM(2,L) = MEDM(L)
 200     CONTINUE
      ENDIF

   ELSE

C  NORMAL VARIATION IS DESIRED

C  CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION

      CALL MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO,
     &              SIGMA2, MCHAT, MU, SIG)

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

   IF (MPROC .EQ. 1) THEN
      CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
   ENDIF

C  COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M

      CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

      CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO,
     &               MPNT, MO, SIGMA2)

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

      CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

      WRITE(7,975)

      DO 350 L = 1, NUMREG
         WRITE(7,980) L, MCHAT(1,L)
 350  CONTINUE

      IF (CZERO .GT. 0.0) THEN
         WRITE(7,960)
         DO 360 L = 1, NUMREG
            IF (MCPNT(L) .EQ. 1) THEN
               WRITE(7,965) L, MC(1,L)
            ELSEIF (MCPNT(L) .EQ. 2) THEN
               WRITE(7,970) L, MC(1,L), MC(2,L)
            ENDIF
 360     CONTINUE
      ENDIF

      WRITE(7,920)
      WRITE(7,930)

      DO 370 L = 1, NUMREG
         WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
 370  CONTINUE

      WRITE(7,950)

      WRITE(7,985)
      DO 380 L = 1, NUMREG
         WRITE(7,990) L, MU(L), SIG(L)
 380  CONTINUE

   ENDIF
```

```
C  PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS

      IF (MPROC .EQ. 1) THEN
         WRITE(7,995) PVAR
      ENDIF

C  FORMAT STATEMENTS

  900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &        'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &        'NASA Contract NAS7-918 is acknowledged.',/////,
     &        2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
     &        ///,2X,'95% CONFIDENCE INTERVALS ON C AND m ',
     &        'FOR EACH REGION',/)

  905 FORMAT(7X,'REGION:   ',I1,7X,'Io = (',F12.9,',',F12.9,')',
     &        /,24X,'Jo = (',F12.9,',',F12.9,')')

  910 FORMAT(//,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
     &        //,7X,'REGION',8X,'E(C)',12X,'E(m)',/)

  915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)

  920 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON m FOR EACH '
     &        'REGION')

  930 FORMAT(//,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

  940 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

  950 FORMAT(///)

  960 FORMAT(//,2X,'RANGE ON m FOR EACH REGION IMPLIED BY C '
     &        'CONSTRAINT',
     &        //,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

  965 FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')

  970 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

  975 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &        'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &        'NASA Contract NAS7-918 is acknowledged.',/////,
     &        2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
     &        ///,2X,'ESTIMATE OF m FOR EACH REGION',
     &        //,7X,'REGION',12X,'E(m)',/)

  980 FORMAT(9X,I1,11X,F10.6)

  985 FORMAT(2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS',
     &        //,2X,'REGION',5X,'MEAN',8X,'STD DEV',/)

  990 FORMAT(5X,I1,5X,F7.4,5X,E11.5)

  995 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ',
     &        'MEDIAN S/N CURVE',/,2X,'WARRANTED BY THE AVAILABLE ',
     &        'INFORMATION',//,7X,E11.5)

      RETURN
      END

C***********************************************************************

C  SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C  ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  5OCT87
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
```

```
C                     V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRMNAT

      WRITE(8,*) 'PROGRAM EXECUTION TERMINATED'
      STOP
      END


C*************************************************************************


C  SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C  USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR,
     &                 REFNP, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

C  INPUTS:    --
C  OUTPUTS:   NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
C             NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
     &        NPTS(0:MAXSET), REFNP(MAXREG)

      REAL    DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
     &        RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG)

C                  LIST OF VARIABLES
C
C  DELTA()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C               SIG() CALCULATION
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  J          CONTROLS DO LOOP FOR EACH DATA SET
C  K          CONTROLS DO LOOP FOR EACH POINT IN A REGION
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LNNF()     3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()    3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET     MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MO()       1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C               MEAN FOR EACH REGION
C  MPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MZERO() FOR EACH REGION
C  MZERO()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C               EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C               IS THE UPPER BOUND
C  NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C               SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP()       2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C               IN EACH REGION
C  NPTS()     1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  RATSTR()   2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
C               STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
```

```
C  RAWNF()      2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                 DATA SETS
C  RAWSTR()     2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OF TOTAL STRAIN
C                 DATA (%) FOR ALL S/N DATA SETS
C  REFNP()      1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C  SIGMA2()     1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 VARIANCE FOR EACH REGION
C  STR()        2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

      DO 100 J = 0, MAXSET
         NPTS(J) = 0.0
  100 CONTINUE

      DO 200 L = 1, MAXREG
         DO 250 J = 0, MAXSET
            NP(J, L) = 0.0
  250    CONTINUE
  200 CONTINUE

      DO 300 J = 0, MAXSET
         DO 350 I = 1, MAXDAT
            RAWNF(I,J) = 0.0
            RAWSTR(I,J) = 0.0
            RATSTR(I,J) = 0.0
  350    CONTINUE
  300 CONTINUE

      DO 400 L = 1, MAXREG
         DO 425 K = 1, MAXDAT
            DO 450 J = 0, MAXSET
               LNNF(K,J,L) = 0.0
               LNSTR(K,J,L) = 0.0
  450       CONTINUE
  425    CONTINUE
  400 CONTINUE

      DO 500 L = 1, MAXREG
         DO 550 K = 1, MAXDAT
           NF(K,L) = 0.0
           STR(K,L) = 0.0
  550    CONTINUE
  500 CONTINUE

      DO 600 L = 1, MAXREG
         REFNP(L) = 0
         MPNT(L) = 0
         MZERO(1,L) = 0.0
         MZERO(2,L) = 0.0
         DELTA(L) = 0.0
         MO(L) = 0.0
         SIGMA2(L) = 0.0
  600 CONTINUE

      RETURN
      END


C**************************************************************************


C  SUBROUTINE RCE "READS" THE DATA FROM SPECFD AND RELATD; "CONVERTS"
C  THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO
C  SPECFO AND RELATO.  RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS
C  SPECIFIED BY USER
C  PROGRAMMER:  L. NEWLIN
C        DATE:  21JUN88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP,
     &                LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
```

```
     &                 NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
     &                 FTUZ, FTYZ, DELTA, MO, SIGMA2, KRATIO, LAMN)
C  INPUTS:    VARY, MPROC
C  OUTPUTS:   NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP,
C             STR, NF, SZERO, ZROREG, NUMREG, NNODAT, NSETS, NBND,
C             CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2,
C             KRATIO, LAMN
C  SUBPROGRAMS:  TRMNAT, CONVRT
C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON IOUT

      INTEGER COUNT, I, IOUT, J, K, L, M, MPNT(MAXREG), MPROC, NDIV,
     &        NNODAT, NP(0:MAXSET, MAXREG), NPTS(0:MAXSET), NSETS,
     &        NUM, NUMREG, REFNP(MAXREG), REG, VARY, ZROREG

      REAL    CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTYZ,
     &        KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        RATIO, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG), SZERO

      CHARACTER*40  DESCRP(0:MAXSET)

C                        LIST OF VARIABLES
C
C  COUNT       INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO,
C                CONVERSION, AND BREAK UP
C  CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                SIG() CALCULATION
C  DESCRP()    1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
C  FTU         ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
C  FTUZ        ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTY         YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  I           CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT        OUTPUT DUMP CONTROLLER
C  J           CONTROLS DO LOOP FOR EACH DATA SET
C  K           CONTROLS DO LOOP FOR EACH POINT IN A REGION
C  KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LAMN        LAMBDA-N - RATIO OF Var (Ln N given S) / (m**2 C**2),
C                CONSTANT OVER ALL REGIONS AND COMPONENTS
C  LNNF()      3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()     3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  M           CONTROLS DO LOOP FOR EACH DATA DIVISION
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C  MPROC       Materials PROCess variation - CONTROLS MATERIALS PROCESS
C                VARIATION - 0 - NO VARIATION;  1 - VARIATION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C  NDIV        NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO,
C                REGION PAIRS DURING INPUT
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
```


C-4

```
C  NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                IN EACH REGION
C  NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUM         NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  RATIO       STRESS RATIO (R = -1.0 IS DESIRED)
C  RATSTR()    2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
C                RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C  RAWNF()     2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                DATA SETS
C  RAWSTR()    2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
C                DATA (%) FOR ALL S/N DATA SETS
C  REFNP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C  REG         REGION OF INTEREST IN A PARTICULAR DIVISION
C  SIGMA2()    1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                VARIANCE FOR EACH REGION
C  STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SZERO       STRESS TENSILE TEST POINT, So
C  VARY        CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO
C                VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                VARIATION;  3 - TRUNCATED NORMAL VARIATION
C  ZROREG      ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                REGION

C  INITIALIZE COUNT AND NBND()

      COUNT = 0

      DO 10 L = 0, MAXREG
         NBND(L) = 0.0
   10 CONTINUE

C  INPUT DATA ON SPECIFIC MATERIAL FROM SPECFD AND ECHO TO SPECFO

      READ(1,*) DESCRP(0), FTY, FTU, NDIV, NPTS(0)

      IF (NPTS(0) .GT. MAXDAT) THEN
         WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &              'SPECIFIC MATERIAL'
         CALL TRMNAT
      ENDIF

      WRITE(3,900) DESCRP(0), FTY, FTU, NPTS(0)
      IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)

      WRITE(3,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

C     STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ

      FTUZ = FTU
      FTYZ = FTY

C     INPUT STRESS/LIFE INFORMATION - INCLUDING STRESS RATIO AND REGION
C     INFORMATION FROM SPECFD AND ECHO TO SPECFO

      DO 100 M = 1, NDIV

         READ (1,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE FOR RATIO:  ', RATIO
            CALL TRMNAT
         ENDIF

         IF (REG .GT. MAXREG) THEN
            WRITE(8,*) 'ERROR:  OVER REGION LIMIT IN SPECIFIC DATA SET'
            CALL TRMNAT
         ENDIF
```

```
          DO 110 I = (COUNT + 1), (COUNT + NUM)
             READ(1,*) RAWSTR(I,0), RAWNF(I,0)
  110     CONTINUE

C         CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT

          IF (RATIO .EQ. -1.0) THEN

C            STRESS RATIO IS CORRECT

             DO 120 I = (COUNT + 1), (COUNT + NUM)
                RATSTR(I,0) = RAWSTR(I,0)
  120        CONTINUE

          ELSE

C            STRESS RATIO TRANSFORMATION MUST BE DONE

             CALL CONVRT (0, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR,
     &                    RATIO, FTU, FTY)

          ENDIF

C         ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL

          DO 130 I = (COUNT + 1), (COUNT + NUM)

             WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
     &                    RATSTR(I,0), RAWNF(I,0)

             IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
     &                    RATIO, REG, RATSTR(I,0), RAWNF(I,0)

  130     CONTINUE

C         BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C         EXPCTD, AND PAREST

          K = NP(0,REG)

          DO 140 I = (COUNT + 1), (COUNT + NUM)

             K = K + 1
             LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
             LNNF(K,0,REG) = ALOG(RAWNF(I,0))
             STR(K,REG) = RATSTR(I,0)
             NF(K,REG) = RAWNF(I,0)

  140     CONTINUE

          IF (K .GT. MAXDAT) THEN
             WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &                  'SPECIFIC MATERIAL'
             CALL TRMNAT
          ENDIF

          NP(0,REG) = K
          REFNP(REG) = K
          COUNT = COUNT + NUM

  100 CONTINUE

      IF (NPTS(0) .NE. COUNT) THEN
         WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &              'INCORRECTLY SPECIFIED'
         WRITE(8,*) 'IN SPECIFIC DATA SET'
         CALL TRMNAT
      ENDIF

      READ(1,*) SZERO
      IF (NINT (SZERO) .GT. 0) THEN
         ZROREG = 0
      ELSE
         ZROREG = 1
```

```
      ENDIF
      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'SZERO = ', SZERO, ' ZROREG = ', ZROREG
C     INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION
      READ(1,*) NUMREG, NNODAT

      IF ((NUMREG + NNODAT) .GT. MAXREG) THEN
         WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON NUMBER OF REGIONS'
         CALL TRMNAT
      ENDIF

      DO 150 L = ZROREG, (NUMREG + NNODAT)
         READ(1,*) NBND(L)
  150 CONTINUE

      READ(1,*) CZERO

      DO 160 L = 1, (NUMREG + NNODAT)
         READ(1,*) MPNT(L), MZERO(1,L), MZERO(2,L)
  160 CONTINUE

      WRITE(3,913)
      IF (ZROREG .EQ. 0) WRITE(3,914) SZERO
      IF (IOUT .EQ. 10) THEN
         WRITE(8,913)
         IF (ZROREG .EQ. 0) WRITE(8,914) SZERO
      ENDIF

      WRITE(3,915) NUMREG, NNODAT
      IF (IOUT .EQ. 10) WRITE(8,915) NUMREG, NNODAT

      DO 170 L = ZROREG, (NUMREG + NNODAT)
         WRITE(3,920) NBND(L)
         IF (IOUT .EQ. 10) WRITE(8,920) NBND(L)
  170 CONTINUE

      WRITE(3,925) CZERO
      IF (IOUT .EQ. 10) WRITE(8,925) CZERO

      DO 180 L = 1, (NUMREG + NNODAT)
         WRITE(3,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
         IF (IOUT .EQ. 10)
     &      WRITE(8,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
         IF ((VARY .EQ. 3) .AND. (MPNT(L) .EQ. 0)) THEN
            WRITE(8,*) 'ERROR:  NORMAL VARIATION REQUIRES A PRIOR ',
     &                 'RANGE ON M'
            CALL TRMNAT
         ENDIF
  180 CONTINUE

      IF (VARY .EQ. 3) THEN
C        READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
         WRITE(3,945)
         IF (IOUT .EQ. 10) WRITE(8,945)
         DO 190 L = 1, (NUMREG + NNODAT)
            READ(1,*) DELTA(L), MO(L), SIGMA2(L)
            WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
            IF (IOUT .EQ. 10)
     &         WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
            IF ((DELTA(L) .LT. 0.0) .OR.
     &         ((DELTA(L) .GT. 0.0) .AND. (MO(L) .LE. 0.0))) THEN
               WRITE(8,*) 'ERROR:  BAD VALUE FOR DELTA OR VALUE OF MO ',
     &                    'INCONSISTENT WITH DELTA IN REGION ', L
               CALL TRMNAT
            ENDIF
  190    CONTINUE
      ENDIF

      IF (MPROC .EQ. 1) THEN
         READ(1,*) KRATIO, LAMN
         WRITE(3,955) KRATIO, LAMN
         IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
      ENDIF
```

```
C  BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATD
C  AND THEN ECHO TO RELATO

      READ(5,*) NSETS

      IF (NSETS .GT. MAXSET) THEN
         WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF RELATED DATA SETS'
         CALL TRMNAT
      ENDIF

      WRITE(6,935) NSETS

      DO 200 J = 1, NSETS

         COUNT = 0

         IF (IOUT. EQ. 10) WRITE(8,*)'J =', J, ' NSETS =', NSETS

         READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)

         IF (NPTS(J) .GT. MAXDAT) THEN
            WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF POINTS IN ',
     &                 'SET ', J
            CALL TRMNAT
         ENDIF

         WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
         IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)

         WRITE(6,905)
         IF (IOUT .EQ. 10) WRITE(8,905)

         DO 300 M = 1, NDIV

            READ(5,*) NUM, RATIO, REG

            IF (ABS(RATIO) .GT. 1.0) THEN
               WRITE(8,*) 'ERROR:  INVALID VALUE OF RATIO:  ', RATIO
               CALL TRMNAT
            ENDIF

            IF (REG .GT. MAXREG) THEN
               WRITE(8,*)
     &            'ERROR:  OVER REGION LIMIT IN RELATED MATERIAL ', J
               CALL TRMNAT
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
               WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
            ENDIF

            DO 310 I = (COUNT + 1), (COUNT + NUM)
               READ(5,*) RAWSTR(I,J), RAWNF(I,J)
  310       CONTINUE

C  CHECK IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT

            IF (RATIO .EQ. -1.0) THEN

C              STRESS RATIO IS CORRECT

               DO 320 I = (COUNT + 1), (COUNT + NUM)
                  RATSTR(I,J) = RAWSTR(I,J)
  320          CONTINUE

            ELSE

C              STRESS RATIO TRANSFORMATION MUST BE DONE
               CALL CONVRT(J, (COUNT + 1), (COUNT + NUM), RAWSTR,
     &                     RATSTR, RATIO, FTU, FTY)

            ENDIF
```

```
C            RECORD BOTH S/N DATA SETS TO RELATO

             DO 330 I = (COUNT + 1), (COUNT + NUM)

                WRITE(6,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG,
     &                       RATSTR(I,J), RAWNF(I,J)

                IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,J), RAWNF(I,J),
     &                       RATIO, REG, RATSTR(I,J), RAWNF(I,J)
 330         CONTINUE

             K = NP(J,REG)

             DO 340 I = (COUNT + 1), (COUNT + NUM)

                K = K + 1
                LNSTR(K,J,REG) = ALOG(RATSTR(I,J))
                LNNF(K,J,REG) = ALOG(RAWNF(I,J))
 340         CONTINUE

             IF (K .GT. MAXDAT) THEN
                WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF POINTS ',
     &                     'IN SET ', J
                CALL TRMNAT
             ENDIF

             NP(J,REG) = K
             COUNT = COUNT + NUM
 300      CONTINUE

          IF (NPTS(J) .NE. COUNT) THEN
             WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &                  'INCORRECTLY SPECIFIED IN SET ', J
             CALL TRMNAT
          ENDIF

 200  CONTINUE


C  FORMAT STATEMENTS USED TO WRITE TO SPECFO AND RELATO

 900  FORMAT(////,13X,'MATERIAL INPUT',///,2X,'DESCRIPTION:',2X,A40,//,
     &        2X,'YIELD STRENGTH',18X,E11.5,//,2X,'ULTIMATE STRENGTH',
     &        15X,E11.5,//,2X,'NUMBER OF POINTS',16X,I2)

 905  FORMAT(//,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
     &        /,5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
     &        'STRESS',7X,'LIFE'/)

 910  FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)

 913  FORMAT(//)

 914  FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN So OF',
     &        5X,E11.5)

 915  FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA   ',
     &        /,2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
     &        /,2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
     &        '(CYCLES): ',/)

 920  FORMAT(10X,E9.3)

 925  FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
     &        'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
     &        //,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
     &        //,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',
     &        5X,'UPPER BOUND',/)

 930  FORMAT(6X,I1,11X,I1,12X,F7.4,9X,F7.4)
```

```
  935 FORMAT(20X,'NUMBER OF DATA SETS:',2X,I2,//,17X,
     &        'NOTE:  ALL Kt ASSUMED TO BE 1.0',////,23X,
     &        'TRANSFORMED DATA')

  940 FORMAT(///,2X,'DESCRIPTION:',2X,A40,
     &        //,2X,'YIELD STRENGTH',18X,F7.0,
     &        //,2X,'ULTIMATE STRENGTH',15X,F7.0,
     &        //,2X,'NUMBER OF POINTS',16X,I2)

  945 FORMAT(/,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS:',
     &        //,2X,'REGION',5X,'DELTA',8X,'mo',10X,'SIGMA2',/)

  950 FORMAT(5X,I1,5X,F7.2,5X,F7.4,5X,E11.5)

  955 FORMAT(//,2X,'MATERIALS PROCESS VARIATION INFORMATION',
     &        //,2X,'MEDK*/MEDK:',5X,E11.5,/,5X,'LAMBDAN:',5X,E11.5)

      RETURN
      END

C***********************************************************************

C  THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C  STRESS RATIO, R, IS NOT -1.0
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87     COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)

C  INPUTS:   J, NUM1, NUM2, STR, R, FTU, FTY
C  OUTPUTS:  RSTR

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXSET

      PARAMETER (MAXDAT = 50, MAXSET = 5)

      COMMON IOUT

      INTEGER I, IOUT, J, NUM1, NUM2

      REAL     FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET),
     &         STR(MAXDAT, 0:MAXSET), TEST

C                  LIST OF VARIABLES
C
C  FTU         ULTIMATE STRENGTH OF MATERIAL (PSI)
C  FTY         YIELD STRENGTH OF MATERIAL (PSI)
C  I           CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C  IOUT        OUTPUT DUMP CONTROLLER
C  J           DATA SET OF INTEREST
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  NUM1        FIRST INDEX TO BE TRANSFORMED
C  NUM2        LAST INDEX TO BE TRANSFORMED
C  R           STRESS RATIO (R = -1.0 IS DESIRED)
C  RSTR()      STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
C  STR()       ARRAY CONTAINING STRESS VALUES (PSI) FOR S/N CURVE
C  TEST        Kt * Smax * (1 - R)/2 , TO BE COMPARED WITH FTY

C  Kt IS ASSUMED TO BE ONE

      DO 100 I = NUM1, NUM2
```

```
          TEST = STR(I,J) * (1.0 - R)/2.0
          IF (IOUT.EQ.10) WRITE(8,*) 'I =',I,' J =',J,' TEST =',TEST

          IF (TEST .GE. FTY) THEN
             RSTR(I,J) = TEST
             IF (IOUT.EQ.10) WRITE(8,*)'1:RSTR() =',RSTR(I,J)
          ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN
             RSTR(I,J) = TEST/(1.0 -((FTY - TEST)/FTU))
             IF (IOUT.EQ.10) WRITE(8,*)'2:RSTR() =',RSTR(I,J)
          ELSE
             RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
     &                   /(2.0 * FTU)))
             IF (IOUT.EQ.10) WRITE(8,*)'3:RSTR() =',RSTR(I,J)
          END IF
  100 CONTINUE
      RETURN
      END

C***********************************************************************

C   SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C   AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C   X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY,
     &                   SY2, DD, SWHAT2, SUHAT2, NPPR)
C   INPUTS:   NUMREG, NSETS, NP, LNSTR, LNNF
C   OUTPUTS:  SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR
C     IMPLICIT NONE
      INTEGER MAXDAT, MAXREG, MAXSET
      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
      COMMON  IOUT
      INTEGER IOUT, J, K, L, NP(0:MAXSET, MAXREG), NPPR(MAXREG),
     &        NSETS, NUMREG
      REAL    BB(MAXREG), DD(MAXREG), DIFFX(MAXDAT, 0:MAXSET),
     &        DIFFY(MAXDAT, 0:MAXSET), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MEANX(0:MAXSET),
     &        MEANY(0:MAXSET), SUHAT2(MAXREG), SWHAT2(MAXREG),
     &        SX2(MAXREG), SXY(MAXREG), SY2(MAXREG)

C                      LIST OF VARIABLES
C
C  BB()       1-D ARRAY CONTAINING SXY(L)/SY2(L) FOR EACH REGION
C  DD()       1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DIFFX()    2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L)
```

```
C                 AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C   DIFFY()     2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L)
C                 AND MEANY(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C   IOUT        OUTPUT DUMP CONTROLLER
C   J           CONTROLS DO LOOP FOR EACH DATA SET
C   K           CONTROLS DO LOOP FOR EACH POINT IN A REGION
C   L           CONTROLS DO LOOP FOR EACH REGION
C   LNNF()      3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C   LNSTR()     3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C   MAXDAT      MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C   MEANX()     1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION
C                L AND DATA SET J (X = Ln S)
C   MEANY()     1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
C                L AND DATA SET J (Y = Ln N)
C   NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                SET IN EACH REGION
C   NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                ALL DATA SETS IN A REGION (Number of Points Per Region)
C   NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   SUHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C   SWHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C   SX2()       1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                (X = Ln S)
C   SXY()       1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
C                EACH REGION (X = Ln S, Y = Ln N)
C   SY2()       1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                (Y = Ln N)

C   INITIALIZE ARRAYS

      DO 50 L = 1, MAXREG
        SY2(L) = 0.0
        SX2(L) = 0.0
        SXY(L) = 0.0
        SWHAT2(L) = 0.0
        SUHAT2(L) = 0.0
        BB(L) = 0.0
        DD(L) = 0.0
        NPPR(L) = 0
   50 CONTINUE

      DO 60 J = 0, MAXSET
         DO 70 K = 1, MAXDAT
            DIFFY(K,J) = 0.0
            DIFFX(K,J) = 0.0
   70    CONTINUE
         MEANY(J) = 0.0
         MEANX(J) = 0.0
   60 CONTINUE

C   NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION

      DO 100 L = 1, NUMREG

         DO 200 J = 0, NSETS
C           FIRST CALCULATE SAMPLE X AND Y MEANS
C           FOR DATA SET J IN REGION L
            MEANY(J) = 0.0
            MEANX(J) = 0.0
            IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' J =', J,
     &                          ' NP =', NP(J,L)

            DO 250 K = 1, NP(J,L)
               MEANY(J) = MEANY(J) + LNNF(K,J,L)
               MEANX(J) = MEANX(J) + LNSTR(K,J,L)
               IF (IOUT .EQ. 10) WRITE(8,*) 'LNNF =', LNNF(K,J,L),
     &              ' LNSTR =', LNSTR(K,J,L)
  250       CONTINUE
```

```
            MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
            MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
            IF (IOUT .EQ. 10) WRITE(8,*) 'MEANY(J) =', MEANY(J),
     &         ' MEANX(J) =', MEANX(J)

C
C           NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY,
C           OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
C           DATA SET IN REGION L

            DO 300 K = 1, NP(J,L)
               DIFFY(K,J) = LNNF(K,J,L) - MEANY(J)
               DIFFX(K,J) = LNSTR(K,J,L) - MEANX(J)
               SY2(L) = SY2(L) + DIFFY(K,J) ** 2
               SX2(L) = SX2(L) + DIFFX(K,J) ** 2
               SXY(L) = SXY(L) + DIFFX(K,J) * DIFFY(K,J)
               IF (IOUT .EQ. 10) THEN
                  WRITE(8,*) 'K =', K, ' DIFFY(K,J) =', DIFFY(K,J),
     &                       ' DIFFX(K,J) =', DIFFX(K,J)
                  WRITE(8,*) 'SY2(L) =', SY2(L), ' SX2(L) =', SX2(L),
     &                       ' SXY(L) =', SXY(L)
               ENDIF
  300       CONTINUE

            NPPR(L) = NPPR(L) + NP(J,L) - 1
            IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) =', NPPR(L)
  200    CONTINUE

         IF (SXY(L) .GE. 0.0) THEN
C           LIFE WILL INCREASE WITH INCREASING STRESS — INVALID FOR
C           OUR MODEL
            WRITE(8,*) 'ERROR: SXY >= 0 IN REGION', L
            CALL TRMNAT
         ENDIF

         NPPR(L) = NPPR(L) - 1

         IF (NPPR(L) .LE. 0) THEN
            WRITE(8,*) 'ERROR:  TOO FEW POINTS FOR REGRESSION IN ',
     &                 'REGION ',L
            CALL TRMNAT
         ENDIF

         SY2(L) = SY2(L) / FLOAT(NPPR(L))
         SX2(L) = SX2(L) / FLOAT(NPPR(L))
         SXY(L) = SXY(L) / FLOAT(NPPR(L))

C        NOW CALCULATE THE RESIDUAL VARIANCES, SWHAT2, SUHAT2, FOR EACH
C        REGION FROM THE Y ON X AND X ON Y REGRESSIONS

         DD(L) = SXY(L) / SX2(L)
         BB(L) = SXY(L) / SY2(L)
         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
     &                 ' SX2(L) =', SX2(L)
            WRITE(8,*) 'SXY(L) =', SXY(L), ' DD(L) =', DD(L),
     &                 ' BB(L) =', BB(L)
         ENDIF

         DO 400 J = 0, NSETS
            IF (IOUT .EQ. 10) WRITE(8,*) 'J =', J, ' NP(J,L) =', NP(J,L)

            DO 500 K = 1, NP(J,L)
               SWHAT2(L) = SWHAT2(L)
     &                   + (DIFFY(K,J) - DD(L) * DIFFX(K,J)) ** 2
               SUHAT2(L) = SUHAT2(L)
     &                   + (DIFFX(K,J) - BB(L) * DIFFY(K,J)) ** 2
               IF (IOUT .EQ. 10) WRITE(8,*) 'K =', K, ' SWHAT2(L) =',
     &            SWHAT2(L), ' SUHAT2(L) =', SUHAT2(L)
  500       CONTINUE

  400    CONTINUE

         SWHAT2(L) = SWHAT2(L) / FLOAT(NPPR(L))
         SUHAT2(L) = SUHAT2(L) / FLOAT(NPPR(L))
         IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) =', NPPR(L),
```

```
     &        ' SWHAT2(L) =', SWHAT2(L), ' SUHAT2(L) =', SUHAT2(L)
  100 CONTINUE

      RETURN
      END


C*********************************************************************


C   SUBROUTINE INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, Io, ON
C   C; AND THE 95% CONFIDENCE INTERVAL, Jo, ON M
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  5OCT87    COMMENTS:  15SEP89
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                   JZERO, MCHAT)

C   INPUTS:    NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C   OUTPUTS:   IZERO, JZERO, MCHAT
C   SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER CHITAB, MAXREG, TTAB

      PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)

      COMMON  IOUT

      INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG

      REAL     ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
     &         IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
     &         SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,
     &         SX2(MAXREG), T, T025(TTAB)

      DATA (CHI025(I), I = 1, 75) /
     &              0.000982069, 0.506356, 0.215795, 0.484419, 0.831211,
     &              1.237347, 1.68987, 2.17973, 2.70039, 3.24697,
     &              3.81575,  4.40379, 5.00874, 5.62872, 6.26214,
     &              6.90766,  7.56418, 8.23075, 8.90655, 9.59083,
     &              10.28293, 10.9823, 11.6885, 12.4011, 13.1197,
     &              13.8439,  14.5733, 15.3079, 16.0471, 16.7908,
     &              17.53,    18.28,   19.04,   19.80,   20.56,
     &              21.33,    22.10,   22.87,   23.65,   24.4331,
     &              25.21,    25.99,   26.78,   27.57,   28.36,
     &              29.15,    29.95,   30.75,   31.55,   32.3574,
     &              33.16,    33.96,   34.77,   35.58,   36.39,
     &              37.21,    38.02,   38.84,   39.66,   40.4817,
     &              41.30,    42.12,   42.95,   43.77,   44.60,
     &              45.43,    46.26,   47.09,   47.92,   48.7576,
     &              49.59,    50.42,   51.26,   52.10,   52.94 /
      DATA (CHI025(I), I = 76, 150) /
     &              53.78,    54.62,   55.46,   56.30,   57.1532,
     &              57.80,    58.84,   59.69,   60.54,   61.39,
     &              62.24,    63.09,   63.94,   64.79,   65.6466,
     &              66.50,    67.35,   68.21,   69.07,   69.92,
     &              70.78,    71.64,   72.50,   73.36,   74.2219,
     &              75.08,    75.94,   76.80,   77.67,   78.53,
     &              79.40,    80.27,   81.13,   82.00,   82.87,
     &              83.73,    84.60,   85.47,   86.34,   87.21,
     &              88.08,    88.95,   89.83,   90.70,   91.57,
     &              92.45,    93.32,   94.19,   95.07,   95.94,
     &              96.82,    97.70,   98.57,   99.45,   100.33,
     &              101.21,   102.09,  102.97,  103.85,  104.73,
     &              105.61,   106.49,  107.37,  108.25,  109.14,
     &              110.02,   110.90,  111.79,  112.67,  113.56,
     &              114.44,   115.33,  116.21,  117.10,  117.98 /
```

```
      DATA (CHI975(I), I = 1, 75) /
     &                5.02389,   7.37776,  9.34840,  11.1433,  12.8325,
     &                14.4494,   16.0128,  17.5346,  19.0228,  20.4831,
     &                21.9200,   23.3367,  24.7356,  26.1190,  27.4884,
     &                28.8454,   30.1910,  31.5264,  32.8523,  34.1696,
     &                35.4789,   36.7807,  38.0757,  39.3641,  40.6465,
     &                41.9232,   43.1944,  44.4607,  45.7222,  46.9792,
     &                48.23,     49.48,    50.72,    51.96,    53.20,
     &                54.44,     55.67,    56.89,    58.12,    59.3417,
     &                60.56,     61.77,    62.99,    64.20,    65.41,
     &                66.62,     67.82,    69.02,    70.22,    71.4202,
     &                72.61,     73.81,    75.00,    76.19,    77.38,
     &                78.57,     79.75,    80.93,    82.12,    83.2976,
     &                84.48,     85.65,    86.83,    88.00,    89.18,
     &                90.35,     91.52,    92.69,    93.86,    95.0231,
     &                96.19,     97.35,    98.52,    99.68,    100.84 /
      DATA (CHI975(I), I = 76, 150) /
     &                102.00,    103.16,   104.31,   105.47,   106.629,
     &                107.78,    108.94,   110.09,   111.24,   112.39,
     &                113.54,    114.69,   115.84,   116.99,   118.136,
     &                119.28,    120.43,   121.57,   122.72,   123.86,
     &                125.00,    126.14,   127.28,   128.42,   129.561,
     &                130.70,    131.84,   132.98,   134.11,   135.25,
     &                136.38,    137.52,   138.65,   139.79,   140.92,
     &                142.05,    143.18,   144.31,   145.44,   146.57,
     &                147.70,    148.83,   149.96,   151.09,   152.21,
     &                153.34,    154.47,   155.59,   156.72,   157.84,
     &                158.97,    160.09,   161.21,   162.33,   163.46,
     &                164.58,    165.70,   166.82,   167.94,   169.06,
     &                170.18,    171.30,   172.41,   173.53,   174.65,
     &                175.77,    176.88,   178.00,   179.12,   180.23,
     &                181.35,    182.46,   183.58,   184.69,   185.80 /

C   VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:
C
C      1 - 30, 40, 50, 60, 70, 80, 90, 100 — Theil, pp. 718-719
C
C      31-39, 41-49, 51-59, 61-69, 71-79, 81-89, 91-99, 101-150
C         — CALCULATED USING CUBE RULE APPROXIMATION


      DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447,
     &             2.365, 2.306, 2.262, 2.228, 2.201, 2.179,
     &             2.160, 2.145, 2.131, 2.120, 2.110, 2.101,
     &             2.093, 2.086, 2.080, 2.074, 2.069, 2.064,
     &             2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /


C                        LIST OF VARIABLES
C
C   ARG         INTERMEDIATE CALCULATION VARIABLE
C   CHI025()    TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C   CHI975()    TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C   CHITAB      MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
C   DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C   I           CONTROLS LOOP FOR CHI025() AND CHI975()
C   IOUT        OUTPUT DUMP CONTROLLER
C   IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                 FOR EACH REGION
C   JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C   L           CONTROLS DO LOOP FOR EACH REGION
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
C                 MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL
C                 DATA SETS IN A REGION (Number of Points Per Region)
C   NUM         EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   SUHAT       EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
```

```
C  SUHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT        EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SWHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SX           EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
C  SX2()        1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                 (X = Ln S)
C  T            VALUE OF T025() USED IN CALCULATIONS
C  T025()       TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
C  TTAB         MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C  INITIALIZE IZERO, JZERO AND MCHAT

      DO 50 L = 1, MAXREG
         IZERO(1,L) = 0.0
         IZERO(2,L) = 0.0
         JZERO(1,L) = 0.0
         JZERO(2,L) = 0.0
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
   50 CONTINUE

C  CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

      DO 75 L = 1, NUMREG
         IF (NPPR(L) .GT. CHITAB) THEN
            WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON DEGREES OF FREEDOM ',
     &                 'IN CHI-SQUARE TABLE, IN REGION ', L
            CALL TRMNAT
         ENDIF
   75 CONTINUE

C  ASSIGN VALUES TO NUM, T, SWHAT, SUHAT AND THEN CALCULATE
C  CONFIDENCE INTERVALS FOR EACH REGION

      DO 100 L = 1, NUMREG

         NUM = NPPR(L)

         IF (NUM .LT. 31) THEN
            T = T025(NUM)
         ELSE
            T = T025(NUM)
         ENDIF

         SWHAT = SWHAT2(L) ** 0.5
         SUHAT = SUHAT2(L) ** 0.5
         SX = (NUM * SX2(L)) ** 0.5

C        CALCULATE ESTIMATED VALUES OF M AND C

         ARG = T * SWHAT / SX
         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SUHAT

C        CALCULATE CONFIDENCE INTERVALS

         IZERO(1,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI975(NUM)) ** 0.5
         IZERO(2,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI025(NUM)) ** 0.5
         JZERO(1,L) = MCHAT(1,L) - ARG
         JZERO(2,L) = MCHAT(1,L) + ARG


         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L =', L, ' NPPR =', NPPR(L), ' NUM =', NUM
            WRITE(8,*) 'SWHAT2 =', SWHAT2(L), ' SWHAT =', SWHAT
            WRITE(8,*) 'SUHAT2 =', SUHAT2(L), ' SUHAT =', SUHAT
            WRITE(8,*) 'SX2 =', SX2(L), ' SX =', SX
            WRITE(8,*) 'CHI025 =', CHI025(NUM), ' CHI975 =', CHI975(NUM)
            WRITE(8,*) 'T =', T, ' DD =', DD(L), ' ARG =', ARG
            WRITE(8,*) 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =',
     &                 IZERO(2,L)
            WRITE(8,*) 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =',
```

```
     &                   JZERO(2,L)
               WRITE(8,*) 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =',
     &                   MCHAT(2,L)
            ENDIF
  100 CONTINUE

      RETURN
      END


C*************************************************************************


C  SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C  THE Co GIVEN BY THE USER
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  8OCT87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

C  INPUTS:   NUMREG, CZERO, SX2, SXY, SY2
C  OUTPUTS:  MCPNT, MC

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, MCPNT(MAXREG), NUMREG

      REAL     ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
     &         SXY(MAXREG), SY2(MAXREG)


C                     LIST OF VARIABLES
C
C  ARG1       INTERMEDIATE CALCULATION VARIABLE
C  ARG2       INTERMEDIATE CALCULATION VARIABLE
C  CZERO      EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C               COEFFICIENT OF VARIATION, Co
C  CZERO2     EQUAL TO CZERO ** 2
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()       2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
C               CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA - MC(1,L) IS
C               THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
C  MCPNT()    1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MC() FOR EACH REGION
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  SX2()      1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C               (X = Ln S)
C  SXY()      1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR
C               EACH REGION (X = Ln S, Y = Ln N)
C  SY2()      1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C               (Y = Ln N)


C     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         MCPNT(L) = 0
         MC(1,L) = 0.0
         MC(2,L) = 0.0
   50 CONTINUE
```

```
C       BEGIN CALCULATIONS

        CZERO2 = CZERO ** 2

        IF (IOUT .EQ. 10)
       &   WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2

        DO 100 L = 1, NUMREG

           ARG1 = SX2(L) - CZERO2
           ARG2 = 0.0

           IF (CZERO .EQ. 0.0) THEN

C             THEN NO M CONSTRAINT IS REQUIRED

              MCPNT(L) = 0

           ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN

C             THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M

              MCPNT(L) = 1
              MC(1,L) = - SY2(L) / (2.0 * SXY(L))

           ELSE

C             THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
C             COMMON CALCULATIONS

              ARG2 = (SXY(L) ** 2 - SY2(L) * ARG1)
              IF (ARG2 .LT. 0.0) THEN
C                ARG2 IS NEGATIVE - IMPLIES M IS COMPLEX
                 WRITE(8,*) 'ERROR:  Co TOO LOW'
                 CALL TRMNAT
              ELSE
                 ARG2 = ARG2 ** 0.5
              ENDIF

              IF (SX2(L) .LT. CZERO2) THEN

C                AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M

                 MCPNT(L) = 1
                 MC(1,L) = (- SXY(L) - ARG2) / ARG1

              ELSE

C                SX2(L) .GT. CZERO2 - THIS TIME THE M CONSTRAINT IS A RANGE

                 MCPNT(L) = 2
                 MC(1,L) = (- SXY(L) - ARG2) / ARG1
                 MC(2,L) = (- SXY(L) + ARG2) / ARG1

              ENDIF

           ENDIF

  100   CONTINUE


        IF (IOUT .EQ. 10) THEN

           DO 200 L = 1, NUMREG

              WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
              WRITE(8,*) 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
              WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)

  200      CONTINUE

        ENDIF
```

```
      RETURN
      END

C***********************************************************************

C  SUBROUTINE GTPVAR CALCULATES THE EXTENT OF DEPARTURE FROM THE MULTIPLE
C  HEAT MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88        COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)

C  INPUTS:    NSETS, NP, NUMREG, LAMN, MCHAT
C  OUTPUTS:   PVAR

C     IMPLICIT NONE

      INTEGER MAXREG, MAXSET

      PARAMETER (MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG),
     &        NUMREG, TOTAL

      REAL    LAMN, MCHAT(2, MAXREG), PSIG2(MAXREG), PVAR, SUM

C
C                      LIST OF VARIABLES
C
C  IOUT           OUTPUT DUMP CONTROLLER
C  J              CONTROLS DO LOOP FOR EACH DATA SET
C  L              CONTROLS DO LOOP FOR EACH REGION
C  LAMN           LAMBDA-N - RATIO OF Var (Ln N given S) / (m**2 C**2),
C                   CONSTANT OVER REGIONS AND COMPONENTS
C  MAXREG         MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET         MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MCHAT()        2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                   FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                   MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
C                   MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  NP()           2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                   SET IN EACH REGION
C  NSETS          NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUM()          EQUAL TO Nj-1 FOR EACH REGION WHERE Nj IS THE SUM OF THE
C                   NUMBER OF POINTS IN EACH DATA SET
C  NUMREG         NUMBER OF REGIONS OF INTEREST
C  PSIG2()        1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
C                   VARIATION IN EACH REGION
C  PVAR           THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C                   CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  SUM            WEIGHTED SUM OF THE PSIG2s - USED TO CALCULATE A WEIGHTED
C                   AVERAGE
C  TOTAL          SUM OF NUM() OVER ALL REGIONS

C     INITIALIZE VARIABLES

      SUM = 0.0
      TOTAL = 0.0

      DO 50 L = 1, MAXREG
         PSIG2(L) = 0.0
         NUM(L) = 0
   50 CONTINUE

      DO 100 L = 1, NUMREG
```

```
        DO 150 J = 0, NSETS
           NUM(L) = NUM(L) + NP(J,L)
 150    CONTINUE
        NUM(L) = NUM(L) - 1
        TOTAL = TOTAL + NUM(L)
 100 CONTINUE

     DO 200 L = 1, NUMREG
        PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
        SUM = SUM + PSIG2(L) * NUM(L)
 200 CONTINUE

     IF (IOUT .EQ. 10) THEN
        WRITE(8,*) 'LAMN = ', LAMN
        DO 300 L = 1, NUMREG
           WRITE(8,*) 'L = ', L, ' NUM = ', NUM(L)
           WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
 300    CONTINUE
        WRITE(8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
     ENDIF

     PVAR = SUM / FLOAT (TOTAL)

     RETURN
     END

C************************************************************************


C  SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
C  M AND Co WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C  TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:   2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

     SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
    &                   MCHAT, RANGEM)

C  INPUTS:    NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:  TRMNAT

C    IMPLICIT NONE

     INTEGER MAXREG

     PARAMETER (MAXREG = 3)

     COMMON  IOUT

     INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

     REAL    JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
    &        MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

C                         LIST OF VARIABLES
C
C  IOUT       OUTPUT DUMP CONTROLLER
C  JZERO()    2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C               FOR EACH REGION
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LOWER      LOWER BOUND OF INTERSECTION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()       2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C               REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C               - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C               BOUND
C  MCHAT()    2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C               FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
```

```
C                 FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MC() FOR EACH REGION
C  MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  UPPER        UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         RANGEM(1,L) = 0.0
         RANGEM(2,L) = 0.0
   50 CONTINUE

C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
         ENDIF

         IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS NO EXOGENOUS INFORMATION
C           ASSUME RANGE TO BE Jo

            RANGEM(1,L) = JZERO(1,L)
            RANGEM(2,L) = JZERO(2,L)

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' JZERO(1,L) = ', JZERO(1,L)
               WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN

C           NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE
C           TO Co, ADJUST THE LOWER BOUND OF Jo ACCORDINGLY

            LOWER = AMAX1(JZERO(1,L), MC(1,L))
            UPPER = JZERO(2,L)
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN

C           THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE
C           CORRESPONDING TO THE Co CONSTRAINT, ADJUST Jo ACCORDINGLY
```

```
         LOWER = AMAX1(JZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF (MPNT(L) .EQ. 1) THEN

C        THERE IS A POINT PRIOR ON M - THIS OVERRIDES ALL OTHER
C        INFORMATION:  ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR

         RANGEM(1,L) = MZERO(1,L)
         RANGEM(2,L) = 0.0

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN

C        THERE IS A PRIOR RANGE ON M, BUT NO Co CONSTRAINT
C        USE INTERSECTION BETWEEN Jo AND Mo

         LOWER = AMAX1(JZERO(1,L), MZERO(1,L))
         UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mo'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN

C        THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C        CONSTRAINT, INTERSECT Jo AND Mo, ADJUSTING THE LOWER BOUND
C        BY Mc ACCORDINGLY

         LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                 'AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF
```

```
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANEGM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN

C           THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO Co CONSTRAINT
C           INTERSECT THESE TWO RANGES WITH Jo

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                    'AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSE

            WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
            CALL TRMNAT

         ENDIF

C        RESTRICT RANGE TO BE NON-NEGATIVE

         RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

         IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)
  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &       .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L
  300 CONTINUE

      RETURN
      END

C***********************************************************************
```

```
C   SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C   WITHOUT DATA
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C      VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C   INPUTS:    RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C   OUTPUTS:   RANGEM, MCHAT, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

      REAL    MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)


C                    LIST OF VARIABLES
C
C   IOUT         OUTPUT DUMP CONTROLLER
C   L            CONTROLS DO LOOP FOR EACH REGION
C   LL           EQUAL TO NUMREG FOR A SET OF CALCULATIONS
C   MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C   MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C                  C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                  MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND
C                  MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                  MZERO() FOR EACH REGION
C   MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                  EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                  IS UPPER BOUND
C   NNODAT       Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NUMREG       NUMBER OF REGIONS OF INTEREST
C   RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                  FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                  RANGEM(2,L) IS THE UPPER BOUND

      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
            ENDIF

C           SPECIFY E(M) OF POSTERIOR FOR SAKE OF
C           CALCULATIONS IN SUBROUTINE EXPCTD

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
            ELSE
```

```
            MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
         ENDIF
         IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL)
      ELSE
         WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &              'SPECIFIED IN REGION WITHOUT DATA'
         CALL TRMNAT
      ENDIF
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE CONCAV ADJUSTS THE UPPER BOUNDS OF THE POSTERIOR CREDIBILITY
C  RANGES ON M TO BE CONSISTENT WITH CONCAVITY CONSTRAINTS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  2FEB88       FORMAT/COMMENTS:  15SEP89
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONCAV (NUMREG, RANGEM)

C  INPUTS:   NUMREG, RANGEM
C  OUTPUTS:  RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    RANGEM(2, MAXREG), TESTM


C                           LIST OF VARIABLES
C
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  RANGEM()   2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C               FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C               RANGEM(2,L) IS THE UPPER BOUND
C  TESTM     UPPER BOUND OF RANGE ON M IN REGION L-1 - USED DURING
C              CONCAVITY ADJUSTMENT


C     ADJUST RANGE TO INSURE CONCAVITY

      DO 100 L = NUMREG, 2, -1

         IF (RANGEM(2,L-1) .EQ. 0.0) THEN
C           RANGE IS A POINT IN REGION L-1
            IF (RANGEM(1,L-1) .GT. AMAX1(RANGEM(1,L),RANGEM(2,L))) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &            ' IS INCONSISTENT WITH POINT POSTERIOR IN REGION ',L-1
               CALL TRMNAT
            ENDIF
         ELSE
C           RANGE IS AN INTERVAL IN REGION L-1
            TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
            IF (TESTM .LT. RANGEM(1,L-1)) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
```

```
     &           ' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
     &           'REGION ', L-1
               CALL TRMNAT
            ELSE
               RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
            ENDIF
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,L-1) =', RANGEM(1,L-1),
     &                 ' RANGEM(2,L-1) =', RANGEM(2,L-1)
            WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                 ' RANGEM(2,L) =', RANGEM(2,L)
            WRITE(8,*) 'TESTM =', TESTM, ' L = ', L
         ENDIF

  100 CONTINUE

      RETURN
      END


C***********************************************************************


C   SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS
C   BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR Co
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  5OCT87      COMMENTS:  1DEC87
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C   INPUTS:    NUMREG, RANGEM
C   IOUTPUT:   MEDM

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG

      REAL     LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)


C                      LIST OF VARIABLES
C
C   IOUT          OUTPUT DUMP CONTROLLER
C   L             CONTROLS DO LOOP FOR EACH REGION
C   LOWERM        LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
C                   TO BE USED IN MEDIAN CALCULATION
C   MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C   MEDM()        1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C   NUMREG        NUMBER OF REGIONS OF INTEREST
C   RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                   FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                   RANGEM(2,L) IS THE UPPER BOUND

C     INITIALIZE ARRAY MEDM

      DO 50 L = 1, MAXREG
         MEDM(L) = 0.0
   50 CONTINUE
```

```
C     BEGIN CALCULATIONS FOR EACH REGION

      DO 100 L = 1, NUMREG

         IF (RANGEM(2,L) .EQ. 0.0) THEN

C           RANGE IS A POINT

            MEDM(L) = RANGEM(1,L)

         ELSEIF (L .EQ. 1) THEN

C           WE ARE IN REGION ONE - NOT AFFECTED BY OTHER REGIONS
C           - MEDIAN WILL JUST BE AVERAGE OF RANGEM VALUES

            MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0

         ELSE

C           MUST TAKE MEDIAN OF REGION L-1 INTO ACCOUNT

            LOWERM = AMAX1(RANGEM(1,L), MEDM(L-1))
            MEDM(L) = (LOWERM + RANGEM(2,L)) / 2.0

         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
            WRITE(8,*) 'LOWERM = ', LOWERM, ' MEDM(L) = ', MEDM(L)
         ENDIF

  100 CONTINUE

      RETURN
      END

C***********************************************************************

C  SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES OF THE S/N
C  CURVE PARAMETERS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  13FEB89       FORMAT/COMMENTS:  15SEP89
C     VERSION:  MATCHR V8.3, V8.4, V8.5    MATGRM V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
     &                   ZROREG, NBND, BIGK1, BZHAT)

C  INPUTS:  NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
C  OUTPUTS:  BIGK1, BZHAT
C  SUBPROGRAMS:  TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG

      REAL      BIGK(0:MAXREG), BIGK1, BZHAT, FACTR, KHAT, MEANZ,
     &          MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
     &          NF(MAXDAT, MAXREG), SBND(0:MAXREG), STR(MAXDAT, MAXREG),
```

```
     &        SZ2, SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)

C                        LIST OF VARIABLES
C
C    BIGK()      1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C                  EACH REGION
C    BIGK1       EQUAL TO BIGK(1)
C    BZHAT       E(BETAo)
C    FACTR       A SCALE FACTOR = PHI * KRATIO * Z
C    IOUT        OUTPUT DUMP CONTROLLER
C    KHAT        E(k)
C    L           CONTROLS DO LOOP FOR EACH REGION
C    MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C    MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C    MEANZ       SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C    MEDM()      1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C    MM()        1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C    NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                  REGIONS OF INTEREST
C    NCOMPS      Number of COMPonents - 1 FOR STRESS AND STRAIN WHEN DECOMPOSED
C                  DATA UNAVAILABLE - 2 FOR DECOMPOSED STRAIN DATA
C    NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                  SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C    NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C                  DATA SET
C    NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR
C                  THE SPECIFIC MATERIAL S/N DATA SET
C    NUMREG      NUMBER OF REGIONS OF INTEREST
C    SBND()      1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C                  CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
C                  CONTAINED IN NBND()
C    STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C                  DATA SET BROKEN INTO REGIONS (PSI OR %)
C    SZ2         SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C    SZERO       STRESS TENSILE TEST POINT, So
C    TRBIGK()    1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
C                  TRBIGK(i) = BIGK(i)
C    ZROREG      ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                  BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C    ZZ()        1-D ARRAY CONTAINING TRANSFORMED S-N DATA, Z = F(STR,NF,NBND,MM)

C    INITIALIZE VARIABLES

        DO 50 L = 0, MAXREG
           MM(L) = 0.0
     50 CONTINUE

C    CREATE MM() ARRAY FROM MEDM() ARRAY

        DO 100 L = 1, NUMREG
           MM(L) = MEDM(L)
    100 CONTINUE

C    TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

        CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C    CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

        CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C    CALCULATE BETAo AND k

        CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

C    CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

        CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

        BIGK1 = BIGK(1)

C    CALCULATE BOUNDARIES OF STRESS REGIONS
```

```
      CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)
C  CALCULATE Ko AND Mo FOR THE NO DATA REGION TO THE LEFT IF REQUIRED

      DO 150 L = ZROREG, NUMREG
         TRBIGK(L) = BIGK(L)
  150 CONTINUE

      IF (ZROREG .EQ. 0) THEN
         FACTR = 1.0
         CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK,
     &                  FACTR, NUMREG)
      ENDIF

C  WRITE RESULTS TO FILE

      IF (NCOMPS .EQ. 1) THEN

         WRITE(7,900) NUMREG, BZHAT, KHAT
         IF (IOUT .EQ. 10) WRITE(8,900) NUMREG, BZHAT, KHAT

         DO 200 L = ZROREG, NUMREG
            WRITE(7,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
            IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
     &                                     NBND(L), SBND(L)
  200    CONTINUE

         WRITE(7,920)

      ELSE

         WRITE(7,930) MM(1), BIGK(1), KHAT

      ENDIF

C  FORMAT STATEMENTS

  900 FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',//,2X,
     &        'NUMBER OF REGIONS:',I4,5X,'E(BETAo) =',F8.4,5X,'E(k) =',
     &        F8.4,//,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',7X,
     &        'STRESS BOUND',/)

  910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E9.3,9X,E11.5)

  920 FORMAT(///)

  930 FORMAT(//,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
     &        //,11X,'m',14X,'K',13X,'E(k)',
     &        //,7X,F8.5,5X,E12.5,6X,F7.4,/)

      RETURN
      END


C***********************************************************************


C  SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C  MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88       COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA,
     &                  MO, SIGMA2, MCHAT, MU, SIG)

C  INPUTS:   NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2
C  OUTPUTS:  MCHAT, MU, SIG

C     IMPLICIT NONE
```

```
      INTEGER   MAXREG

      PARAMETER (MAXREG = 3)

      COMMON    IOUT

      INTEGER   IOUT, L, NUMREG, NPPR(MAXREG)

      REAL      ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG),
     &          MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG),
     &          SUHAT2(MAXREG), SUMX2, SWHAT2(MAXREG), SX2(MAXREG)

C               LIST OF VARIABLES
C
C    ARG       INTERMEDIATE CALCULATION VARIABLE
C    DD()      1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C    DELTA()   1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                SIG() CALCULATION
C    IOUT      OUTPUT DUMP CONTROLLER
C    L         CONTROLS DO LOOP FOR EACH REGION
C    MAXREG    MAXIMUM NUMBER OF REGION ALLOWED
C    MCHAT()   2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR
C                EACH REGION, BASED ON MATERIALS DATA ONLY — MCHAT(1,L) =
C                - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT,
C                THE ESTIMATE FOR C
C    MO()      1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C    MU()      1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C    NPPR()    1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL
C                DATA SETS IN A REGION (Number of Points Per Region)
C    NUMREG    NUMBER OF REGIONS OF INTEREST
C    SIG()     1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C    SIGMA2()  1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                VARIANCE FOR EACH REGION
C    SUHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C    SUMX2     EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
C    SWHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C    SX2()     1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                (X = Ln S)

C     INITALIZE ARRAYS

      DO 50 L = 1, MAXREG
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
         MU(L) = 0.0
         SIG(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATION FOR EACH REGION

      DO 100 L = 1 , NUMREG

         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SQRT (SUHAT2(L))
         SUMX2 = NPPR(L) * SX2(L)
         ARG = SUMX2 + DELTA(L)

         IF (DELTA(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
C           USE THE ESTIMATE OF M
            MU(L) = MCHAT(1,L)
         ELSE
C           UPDATE THE ESTIMATE OF M WITH Mo USING DELTA
            MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
         ENDIF

         IF (SIGMA2(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
```

```
C           USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
            SIG(L) = SQRT (SWHAT2(L) / ARG)
         ELSE
            SIG(L) = SQRT (SIGMA2(L) / ARG)
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ',
     &                  MCHAT(1,L)
            WRITE(8,*) 'SUHAT2 = ', SUHAT2(L),   ' MCHAT2 = ',
     &                  MCHAT(2,L)
            WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
     &                 ' SUMX2 = ', SUMX2
            WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
            WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
            WRITE(8,*) 'SWHAT2 = ', SWHAT2(L), ' SIGMA2 = ', SIGMA2(L),
     &                 ' SIG = ', SIG(L)
         ENDIF

  100 CONTINUE

      RETURN
      END

C***********************************************************************


C  SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND Co TO
C  OBTAIN POSTERIOR RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  10FEB88       FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  INPUTS:    NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
     &        RANGEM(2, MAXREG), UPPER


C                          LIST OF VARIABLES
C
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LOWER      LOWER BOUND OF INTERSECTION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()       2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C               REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C               - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C               BOUND
C  MCHAT()    2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C               FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
C               FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()    1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MC() FOR EACH REGION
C  MPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MZERO() FOR EACH REGION
C  MZERO()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
```

```
C                 EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C   UPPER       UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         RANGEM(1,L) = 0.0
         RANGEM(2,L) = 0.0
   50 CONTINUE

C   PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
         ENDIF

         IF (MPNT(L) .EQ. 1) THEN

C           THERE IS A POINT PRIOR ON M - THIS OVERRIDES ALL OTHER
C           INFORMATION:  ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR

            RANGEM(1,L) = MZERO(1,L)
            RANGEM(2,L) = 0.0

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS A PRIOR RANGE ON M, BUT NO Co CONSTRAINT USE Mo

               RANGEM(1,L) = MZERO(1,L)
               RANGEM(2,L) = MZERO(2,L)

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN

C           THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C           CONSTRAINT ADJUST THE LOWER BOUND OF Mo BY Mc

            LOWER = AMAX1(MZERO(1,L), MC(1,L))
            UPPER = MZERO(2,L)
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANEGM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
```

```
            ENDIF
         ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN
C           THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO Co CONSTRAINT
C           INTERSECT THESE TWO RANGES
            LOWER = AMAX1(MZERO(1,L), MC(1,L))
            UPPER = AMIN1(MZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF
         ELSE
            WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
            CALL TRMNAT
         ENDIF
C        RESTRICT RANGE TO BE NON-NEGATIVE
         RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)
         IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)
  100 CONTINUE
C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE
      DO 300 L = 1, NUMREG
         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &       .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L
  300 CONTINUE
      RETURN
      END
C*************************************************************************
C  SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C  DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  10FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
      SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
     &                   MZERO, MPNT, MO, SIGMA2)
C  INPUTS:   RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
```

```
C             MO, SIGMA2
C   OUTPUTS:  RANGEM, MCHAT, MU, SIG, NUMREG

C      IMPLICIT NONE

       INTEGER MAXREG

       PARAMETER (MAXREG = 3)

       COMMON  IOUT

       INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

       REAL     MCHAT(2, MAXREG), MO(MAXREG), MU(MAXREG),
     &          MZERO(2, MAXREG), RANGEM(2, MAXREG), SIG(MAXREG),
     &          SIGMA2(MAXREG)


C                    LIST OF VARIABLES
C
C   IOUT          OUTPUT DUMP CONTROLLER
C   L             CONTROLS DO LOOP FOR EACH REGION
C   LL            EQUAL TO NUMREG FOR A SET OF CALCULATIONS
C   MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C   MCHAT()       2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C                   C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                   MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND
C                   MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   MO()          1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                   MEAN FOR EACH REGION
C   MPNT()        1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                   MZERO() FOR EACH REGION
C   MU()          1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                   DISTRIBUTION MEAN FOR EACH REGION
C   MZERO()       2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                   EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                   IS UPPER BOUND
C   NNODAT        Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NUMREG        NUMBER OF REGIONS OF INTEREST
C   RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                   FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                   RANGEM(2,L) IS THE UPPER BOUND
C   SIG()         1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                   DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C   SIGMA2()      1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                   VARIANCE FOR EACH REGION


      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            MU(LL) = MO(LL)
            SIG(LL) = SQRT(SIGMA2(LL))
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
               WRITE(8,*) 'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
               WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
     &                    SIGMA2(LL)
            ENDIF

C           SPECIFY E(M) OF POSTERIOR FOR SAKE OF
C           CALCULATIONS IN SUBROUTINE EXPCTD
```

```
            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
               MU(LL) = RANGEM(1,LL)
               SIG(LL) = 0.0
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL),
     &         ' MU = ', MU(LL), ' SIG = ', SIG(LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C   SUBROUTINE PAREST CONTROLS THE CALCULATIONS FOR THE PARAMETER
C   ESTIMATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  13FEB89       FORMAT/COMMENTS:  15SEP89
C      VERSION:  MATCHR V8.3, V8.4, V8.5 - FOR USE WITH PFM'S
C                MATGRM V4.3, V4.4, V4.5
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.


      SUBROUTINE PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG,
     &                   ZROREG, RAND, NBND, STR, BIGK, BZERO, MM,
     &                   SBND)

C  INPUTS:     VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
C              NBND, STR
C  OUTPUTS:    BIGK, BZERO, MM, SBND
C  SUBPROGRAMS:  FINDM, FINDMN, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, VARY, ZROREG

      REAL     BIGK(0:MAXREG), BZERO, K, MEANZ, MM(0:MAXREG),
     &         MU(MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &         RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
     &         STR(MAXDAT, MAXREG), SZ2, ZZ(MAXDAT)

      DOUBLE PRECISION RAND


C                       LIST OF VARIABLES
C
C  BIGK()   1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C             EACH REGION
C  BZERO    VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C  IOUT     OUTPUT DUMP CONTROLLER
C  K        VALUE OF k - PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C  L        CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT   MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG   MAXIMUM NUMBER OF REGIONS ALLOWED
```

```
C  MEANZ     SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MU()      1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C              DISTRIBUTION MEAN FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NF()      2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C              SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP        TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()    1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C              SPECIFIC MATERIAL S/N DATA SET
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  RANGEM()  2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C              FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C              RANGEM(2,L) IS THE UPPER BOUND
C  RAND      RANDOM NUMBER SEED
C  SBND()    1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C              REGION CONTAINED IN NBND()
C  SIG()     1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
C              STANDARD DEVIATION FOR EACH REGION
C  STR()     2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C              DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SZ2       SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  VARY      CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO VARIATION;
C              1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;
C              3 - TRUNCATED NORMAL VARIATION
C  ZROREG    ZeRo REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C  ZZ()      1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C              Z = F(STR,NF,NBND,MM)

C  OBTAIN THE VALUES OF M FOR EACH REGION

      IF (VARY .LE. 2) THEN

C        UNIFORM OR NO VARIATION IN M IS DESIRED

         CALL FINDM (RAND, NUMREG, RANGEM, MM)

      ELSE

C        NORMAL VARIATION IN M IS DESIRED

         CALL FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)

      ENDIF

C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

      CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C  CALCULATE THE VALUES FOR k AND BETAo FROM THE SAMPLE MEAN
C  AND VARIANCE

      CALL KBETA (MEANZ, SZ2, K, BZERO)

C  CALCULATE THE VALUE OF K FOR EACH REGION WHERE A = K ** M

      CALL FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C  CALCULATE STRESS TIE-POINTS

      CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C  WRITE RESULTS TO FILE

C       WRITE(7,900) NUMREG, BZERO

C       DO 200 L = ZROREG, NUMREG
```

```
C          WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
C  200 CONTINUE

C       WRITE(7,920)

C  FORMAT STATEMENTS

  900 FORMAT(///,2X,'SELECTED VALUES OF S/N CURVE PARAMETERS',
     &       //,2X,'NUMBER OF REGIONS: ',I4,5X,'BETAo = ',F8.4,
     &       //,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',5X,
     &       'STRESS BOUND',/)
  910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E9.3,6X,E11.5)
  920 FORMAT(///)

      RETURN
      END

C***********************************************************************

C  SUBROUTINE FINDM CALCULATES THE VALUE OF M FOR EACH REGION BY
C  SAMPLING OFF THE APPROPRIATE M RANGE
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

C  INPUTS:    RAND, NUMREG, RANGEM
C  OUTPUTS:   MM
C  SUBPROGRAMS:  RANDOM, TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL     MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X

      DOUBLE PRECISION RAND

C                         LIST OF VARIABLES
C
C  IOUT          OUTPUT DUMP CONTROLLER
C  L             CONTROLS DO LOOP FOR EACH REGION
C  MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()          1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NUMREG        NUMBER OF REGIONS OF INTEREST
C  PICK()        1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C  RAND          RANDOM NUMBER SEED
C  RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                  FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                  RANGEM(2,L) IS THE UPPER BOUND
C  X             UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                  OFF THE RANGE ON M

C  INITIALIZE MM()

      DO 50 L = 0, MAXREG
         MM(MAXREG) = 0.0
   50 CONTINUE

C  BEGIN CALCULATIONS
```

```
      DO 100 L = 1, NUMREG

         PICK(1) = 0.0
         PICK(2) = 0.0

         IF (RANGEM(2,L) .EQ. 0.0) THEN
C           M IS SPECIFIED AS A POINT VALUE
            MM(L) = RANGEM(1,L)
            IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                                   ' MM(L) =', MM(L)
         ELSEIF (L .EQ. 1) THEN
C           SAMPLE ON EXISTING RANGE
            CALL RANDOM(X, RAND)
            MM(L) = (RANGEM(2,L) - RANGEM(1,L)) * X + RANGEM(1,L)
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                    ' RANGEM(2,L) =', RANGEM(2,L)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS - THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              SAMPLE ON ADJUSTED RANGE
               CALL RANDOM (X, RAND)
               MM(L) = (PICK(2) - PICK(1)) * X + PICK(1)
            ENDIF
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'L =', L, ' MM(L-1) =', MM(L-1),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
         ENDIF

  100 CONTINUE

      RETURN
      END


C************************************************************************


C***********************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C***********************************************************************

      SUBROUTINE RANDOM (FRAC, RAND)
C     IMPLICIT NONE
      COMMON  IOUT
      INTEGER IOUT
```

```
      REAL    FRAC
      DOUBLE PRECISION  RANA, RANC, RAND, RANDIV, RANM, RANSUB,
     &          RANT, RANX

C                  LIST OF VARIABLES
C
C    FRAC      UNIFORM (0,1) RANDOM VARIATE
C    IOUT      OUTPUT DUMP CONTROLLER
C    RANA      CONSTANT FOR LCG
C    RANC      CONSTANT FOR LCG
C    RAND      RANDOM NUMBER SEED
C    RANDIV    INTERNAL CALCULATION
C    RANM      CONSTANT FOR LCG
C    RANSUB    INTERNAL CALCULATION
C    RANT      INTERNAL CALCULATION
C    RANX      INTERNAL CALCULATION


C          USING LCG RANDOM # GENERATOR

      RANA = 671093.0
      RANC = 7090885.0
      RANM = 33554432.0

   10 RANX = RANA * RAND + RANC
      RANDIV = RANX / RANM
      RANT = DINT(RANDIV)
      RANSUB = RANT * RANM
      RAND = RANX - RANSUB
      FRAC = SNGL(RAND / RANM)

      IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
      IF (IOUT .EQ. 2) WRITE(8,*)'RANX =', RANX, ' RANDIV =', RANDIV,
     &    ' RANT =', RANT, ' RANSUB =', RANSUB, ' RAND =', RAND,
     &    ' FRAC =', FRAC

      RETURN
      END

C     NOTES:  IOUT=2 DUMPS TO SCREEN


C************************************************************************


C  SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C  SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13FEB89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)

C  INPUTS:    RAND, NUMREG, MU, SIG, RANGEM
C  OUTPUTS:   MM
C  SUBPROGRAMS:  NORMGN, TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL     MM(0:MAXREG), MU(MAXREG), PICK(2), RANGEM(2, MAXREG),
     &         SIG(MAXREG), X
```

```
      DOUBLE PRECISION RAND

C                           LIST OF VARIABLES
C
C   IOUT          OUTPUT DUMP CONTROLLER
C   L             CONTROLS DO LOOP FOR EACH REGION
C   MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C   MM()          1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C   MU()         1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
C   NUMREG        NUMBER OF REGIONS OF INTEREST
C   PICK()        1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C   RAND          RANDOM NUMBER SEED
C   RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                   FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                   RANGEM(2,L) IS THE UPPER BOUND
C   SIG()        1-D ARRAY CONTAINING THE STANDARD DEVIATION OF M FOR EACH
C                   REGION
C   X             NORMAL(MU,SIGMA) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                   OFF THE RANGE ON M


C   INITIALIZE MM()

      DO 50 L = 0, MAXREG
         MM(MAXREG) = 0.0
   50 CONTINUE

C   BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG

         PICK(1) = 0.0
         PICK(2) = 0.0

         IF (RANGEM(2,L) .EQ. 0.0) THEN
C           M IS SPECIFIED AS A POINT VALUE
            MM(L) = RANGEM(1,L)
            IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                                   ' MM(L) =', MM(L)
         ELSEIF (L .EQ. 1) THEN
C           SAMPLE ON EXISTING RANGE
   10       CALL NORMGN (RAND, MU(L), SIG(L), X)
            IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
            MM(L) = X
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                    ' RANGEM(2,L) =', RANGEM(2,L)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS - THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              SAMPLE ON ADJUSTED RANGE
   20          CALL NORMGN (RAND, MU(L), SIG(L), X)
               IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
               MM(L) = X
            ENDIF
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'L =', L, ' MM(L-1) =', MM(L-1),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
         ENDIF
```

```
  100 CONTINUE

      RETURN
      END

C*********************************************************************


C*********************************************************************
C  SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C  WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  3FEB88
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
C
C  The random variates are generated using the "Direct Method"
C  Abramowitz, M., and Stegun, I. A., editors, Handbook of
C    Mathematical Functions, National Bureau of Standards, Applied
C    Mathematics Series 55, Issued June 1964, Ninth Printing, November
C    1970 with corrections, pg. 953.
C*********************************************************************

      SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C     SUBPROGRAM:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2

      PARAMETER (PI = 3.1415926536)

      INTEGER IOUT

C              LIST OF VARIABLES
C
C  FRAC     UNIFORM(0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  MU       MEAN OF NORMAL DISTRIBUTION
C  RAND     RANDOM NUMBER SEED
C  SIGMA    STANDARD DEVIATION OF NORMAL DISTRIBUTION
C  X        NORMAL RANDOM VARIATE
C  U1       UNIFORM RANDOM NUMBER U(0,1)
C  U2       UNIFORM RANDOM NUMBER U(0,1)
C  Z1       NORMAL RANDOM NUMBER ON N(0,1)
C  Z2       NORMAL RANDOM NUMBER ON N(0,1)

      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'RAND =', RAND, ' MU =', MU, ' SIGMA =', SIGMA

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

      CALL RANDOM (FRAC, RAND)
      U2 = FRAC
      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'U1 =', U1,' U2 =', U2

      Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
      Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)

      X = SIGMA * Z1 + MU
      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'Z1 =', Z1, ' Z2 =', Z2, ' X =', X
```

```
      RETURN
      END

C***********************************************************************

C   SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM
C   THE S/N DATA INTO THE VARIABLE Z = Ln(X)
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  7JUN88      COMMENTS:  13JUL89
C      VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C   INPUTS:    NPTS, STR, NF, NUMREG, MM, NBND
C   OUTPUTS:   NP, ZZ

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON IOUT

      INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG

      REAL     MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &         STR(MAXDAT, MAXREG), ZZ(MAXDAT)

C                    LIST OF VARIABLES
C
C   I          CONTROLS DO LOOP FOR EACH DATA POINT
C   IOUT       OUTPUT DUMP CONTROLLER
C   K          CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
C   L          CONTROLS DO LOOP FOR EACH REGION
C   LL         CONTROLS INNER DO LOOP FOR EACH REGION
C   MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C   MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C   MM()       1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
C   MML        EQUAL TO MM(L) FOR A SET OF CALCULATIONS
C   NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C   NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C   NPTS()     1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C                SPECIFIC MATERIAL S/N DATA SET
C   NUMREG     NUMBER OF REGIONS OF INTEREST
C   STR()      2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
C   ZZ()       1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
C                Z = F(STR,NF,NBND,MM)

C   INITIALIZE VARIABLES

      NP = 0

      DO 50 I = 1, MAXDAT
         ZZ(I) = 0.0
   50 CONTINUE

C   BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG
         MML = MM(L)
```

```
        IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =',
     &    MML, ' NPTS =', NPTS(L)

        DO 200 K = 1, NPTS(L)
           NP = NP + 1
           ZZ(NP) = ALOG(STR(K,L)) + ALOG(NF(K,L)) * (1.0 / MML)
           IF (IOUT .EQ. 10) WRITE(8,*)'K =', K, ' NP =', NP, ' NF =',
     &        NF(K,L), ' STR =', STR(K,L), ' ZZ =', ZZ(NP)

           DO 300 LL = 2, L
              ZZ(NP) = ZZ(NP) + ALOG(NBND(LL-1))
     &                 * ((1.0 / MM(LL-1)) - (1.0 / MM(LL)))
              IF (IOUT .EQ. 10) WRITE(8,*)'LL =', LL, ' NBND(LL-1) =',
     &            NBND(LL-1), ' MM(LL-1) =', MM(LL-1), ' MM(LL) =',
     &            MM(LL), ' ZZ =', ZZ(NP)
 300       CONTINUE

 200    CONTINUE

 100 CONTINUE

      RETURN
      END


C**********************************************************************


C  SUBROUTINE SMNVAR CALCULATES THE Sample MeaN and VARiance OF
C  Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  24AUG87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V3.3, V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE SMNVAR (NP, ZZ, MEANZ, SZ2)

C  INPUTS:    NP, ZZ
C  OUTPUTS:   MEANZ, SZ2

C     IMPLICIT NONE

      INTEGER MAXDAT

      PARAMETER (MAXDAT = 50)

      COMMON IOUT

      INTEGER I, IOUT, NP

      REAL    MEANZ, SZ2, ZZ(MAXDAT)


C                   LIST OF VARIABLES
C
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C               DATA SET
C  SZ2        SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  ZZ()       1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C               Z = F(STR,NF,NBND,MM)


C  INITIALIZE VARIABLES

      MEANZ = 0.0
      SZ2 = 0.0
```

```
C  CALCULATE THE MEAN OF ZZ(), MEANZ

      DO 100 I = 1, NP
         MEANZ = MEANZ + ZZ(I)
         IF (IOUT .EQ. 10) WRITE(8,*)'NP =', NP, ' I =', I,
     &      ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
  100 CONTINUE
      MEANZ = MEANZ / FLOAT(NP)
      IF (IOUT .EQ. 10) WRITE(8,*)' MEANZ =', MEANZ

C  CALCULATE THE VARIANCE OF ZZ(), SZ2

      DO 200 I = 1, NP
         SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
         IF (IOUT .EQ. 10) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
  200 CONTINUE
      SZ2 = SZ2 / FLOAT(NP - 1)
      IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 =', SZ2

      RETURN
      END


C************************************************************************


C  SUBROUTINE KBETA CALCULATES k AND BETAo FROM THE SAMPLE MEAN AND
C  VARIANCE OF Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)

C  INPUTS:   MEANZ, SZ2
C  OUTPUTS:  K, BZERO

C     IMPLICIT NONE

      REAL     PI

      PARAMETER (PI = 3.1415926536)

      COMMON  IOUT

      INTEGER IOUT

      REAL     BZERO, K, MEANZ, SZ, SZ2


C                  LIST OF VARIABLES
C
C  BZERO        VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE
C                 SPECIFIC MATERIAL S/N DATA SET
C  IOUT         OUTPUT DUMP CONTROLLER
C  K            VALUE OF k - PARAMETER CHARACTERIZING SPECIFIC MATERIAL
C                 DATA BASE
C  MEANZ        SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  PI           SELF EXPLANATORY CONSTANT
C  SZ           SZ2 ** 0.5
C  SZ2          SAMPLE VARIANCE OF THE TRANSFORMED DATA,
C                 Z = F(STR, NF, NBND, MM)


C  PERFORM CALCULATIONS


      SZ = SZ2 ** 0.5
```

```
      BZERO = PI / (SZ * (6.0 ** 0.5))

      K = MEANZ

C  DATA DUMP STATEMENTS

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'SZ2 =', SZ2, ' SZ =', SZ
         WRITE(8,*) 'MEANZ =', MEANZ, ' K = ', K, ' BZERO =', BZERO
      ENDIF

      RETURN
      END


C*************************************************************************


C  SUBROUTINE FINDK CALCULATES THE VALUE OF K, WHERE A = K ** M FOR
C  EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  7JUN88
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C  INPUTS:    BZERO, K, MM, NBND, NUMREG
C  OUTPUTS:   BIGK

C     IMPLICIT NONE

      INTEGER MAXREG

      REAL    GAMMA

      PARAMETER (GAMMA = 0.57721566490, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    BIGK(0:MAXREG), BZERO, K, MM(0:MAXREG), NBND(0:MAXREG)


C                     LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC
C              MATERIAL DATA BASE
C  GAMMA     EULER'S CONSTANT
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         VALUE OF k - PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C              DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST

C  INITIALIZE VARIABLES

      DO 50 L = 0, MAXREG
         BIGK(L) = 0.0
   50 CONTINUE

C  CALCULATE K FOR REGION ONE
```

```
      BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
C       WRITE(7,*) 'REGION:  1,  K =', BIGK(1)
      IF (IOUT .EQ. 10) WRITE(8,*)'BZERO =', BZERO, ' k =', K,
     &   ' GAMMA =', GAMMA, ' BIGK(1) =', BIGK(1)

C  CALCULATE K FOR REMAINING REGIONS

      DO 100 L = 2, NUMREG
         BIGK(L) = BIGK(L-1) * NBND(L-1)
     &           ** ((1.0 / MM(L)) - (1.0 / MM(L-1)))
C          WRITE(7,*) 'REGION ', L, ' K =', BIGK(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' NBND(L-1) =',
     &      NBND(L-1), ' MM(L) =', MM(L), ' MM(L-1) =', MM(L-1),
     &      ' BIGK(L) =', BIGK(L)
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE FINDSB CALCULATES THE REGION 'TIE-POINTS' - THE STRESS
C  VALUES WHICH CORRESPOND TO THE "LIFE BOUNDARIES" ACCORDING TO THE
C  RANDOMLY SELECTED Ms, AND THE Ks CALCULATED FROM THE BETA AND k
C  CHARACTERIZING SPECIFIC MATERIAL
C  PROGRAMMER:  L. NEWLIN
C        DATE:  22DEC88
C     VERSION:  MATCHR V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C  INPUTS:    NUMREG, ZROREG, NBND, BIGK, MM
C  OUTPUTS:   SBND

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG, ZROREG

      REAL     BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
     &         SBND(0:MAXREG)


C             LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  IOUT      OUTPUT DUMP CONTROLLER
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  SBND()    1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C              REGION CONTAINED IN NBND()
C  ZROREG    ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO REGION

C  INITIALIZE SBND()

      DO 50 L = 0, MAXREG
         SBND(L) = 0.0
```

```
   50 CONTINUE
C     CALCULATE SBND(0) IF ZROREG = 0
      IF (ZROREG .EQ. 0) THEN
         SBND(0) = BIGK(1) * NBND(0) ** (-1.0 / MM(1))
      ENDIF
C     CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES
      DO 100 L = 1, NUMREG
         IF (NBND(L) .GE. 1.0E+36) THEN
            SBND(L) = 0.0
         ELSE
            SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
         ENDIF
  100 CONTINUE
      RETURN
      END


C***********************************************************************


C  THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
C  MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C  TRANSFORM METHOD"
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  18MAR87      COMMENTS:  15SEP89
C     VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                      V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                      V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE WEIBGN (BETA, RAND, WEIB)
C  INPUTS:   BETA, RAND
C  OUTPUTS:  WEIB
C  SUBPROGRAMS:  RANDOM
C     IMPLICIT NONE
      COMMON  IOUT
      INTEGER IOUT
      REAL    ARG, BETA, ETA, FRAC, WEIB
      DOUBLE PRECISION RAND

C                LIST OF VARIABLES
C
C  ARG      INTERMEDIATE CALCULATION VARIABLE
C  BETA     WEIBULL DISTRIBUTION SHAPE PARAMETER
C  ETA      WEIBULL DISTRIBUTION LOCATION PARAMETER
C  FRAC     UNIFORM (0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  RAND     RANDOM NUMBER SEED
C  WEIB     WEIBULL(BETA,ETA) GENERATED RANDOM VARIATE


C     CALCULATE CONSTRAINED ETA
      ETA = 1.0 / (ALOG(2.0) ** (1.0 / BETA))
```

```
C     GENERATE WEIBULL RANDOM VARIATE

      CALL RANDOM(FRAC, RAND)
      ARG = -ALOG(1.0 - FRAC)
      WEIB = ETA * ARG**(1.0/BETA)
      IF (IOUT .EQ. 10) WRITE(8,*)'BETA = ', BETA, ' ETA =', ETA,
     &    ' FRAC =', FRAC, ' ARG =', ARG, ' WEIB =', WEIB

      RETURN
      END


C***********************************************************************


C  SUBROUTINE KOMO CALCULATES Ko AND Mo FOR THE ZERO REGION (NO DATA
C  REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT
C  AT SZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
C  PROGRAMMER :  L. NEWLIN
C         DATE:  1AUG91
C      VERSION:  MATCHR V8.5   MATGRM V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.


      SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
     &                 FACTR, NUMREG)

C  INPUTS:   SZERO, BIGK, MM, NBND, TRSBND, FACTR
C  OUTPUTS:  TRBIGK, MM, TRSBND

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG

      REAL     BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1         SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)


C                        LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C              EACH REGION
C  FACTR     SCALE FACTOR = PHI * KRATIO * Z
C  IOUT      OUTPUT DUMP CONTROLLER
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS
C  SCLK      ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
C  SZERO     STRESS TENSILE TEST POINT, So
C  TRBIGK()  1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP
C              SBND(0) < So FOR EACH TRIAL
C  TRSBND()  1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE
C              LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()
C              ADJUSTED BY VARIATION PARAMETERS FOR EACH TRIAL


      BIGK(0) = SZERO

      IF (TRSBND(0) .GT. SZERO) THEN
         SCLK = SZERO/TRSBND(0)
         DO 100 L = 0, NUMREG
```

```
            TRBIGK(L) = BIGK(L) * SCLK
            TRSBND(L) = TRSBND(L) * SCLK
100      CONTINUE
      ELSE
         TRBIGK(0) = SZERO/FACTR
         MM(0) = MM(1) * ((ALOG (BIGK(1)) - ALOG (TRSBND(0))
     &           + ALOG (FACTR)) / (ALOG (SZERO) - ALOG (TRSBND(0))))
      ENDIF
C
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'SZERO = ', SZERO, ' BIGKo = ', TRBIGK(0)
         WRITE(8,*) 'FACTOR = ', FACTR, ' BIGK1 = ', TRBIGK(1)
         WRITE(8,*) 'MM1 = ', MM(1), ' MMo = ', MM(0)
      ENDIF

      RETURN
      END

C***********************************************************************

C  FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
C  BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB89
C     VERSION:  MATCHR V8.3, V8.4, V8.5 — FOR USE WITH PFM'S
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      REAL FUNCTION GTLIFE (S, MM, LNA, LPHIM, KRATIO, LNZ, SBND,
     &                      ZROREG, NUMREG, SZERO)

C  INPUTS:   S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO
C  OUTPUTS:  GTLIFE

C     IMPLICIT NONE

      INTEGER IOUT, L, MAXREG, NUMREG, ZROREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      REAL     GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
     &         MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

C
C                         LIST OF VARIABLES
C
C  GETLIF     VALUE TO BE ASSIGNED TO GTLIFE — CYCLES TO FAILURE FOR
C               THE REQUIRED STRESS LEVEL
C  IOUT       OUTPUT DUMP CONTROLLER
C  KRATIO     RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LNA()      1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C  LNZ        NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  LPHIM()    1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
C               PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  S          VALUE OF STRESS (PSI) FOR WHICH A VALUE OF LIFE (CYCLES TO
C               FAILURE) IS REQUIRED
C  SBND()     1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C               CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
C               CONTAINED IN NBND()
C  SZERO      STRESS TENSILE POINT, So
C  TEMP       TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER
C               FLOWS
```

```
C  ZROREG     ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C               BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO REGION

      GETLIF = 0.0

C  CALCULATE CYCLES TO FAILURE

      IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
         GETLIF = 1.0
      ELSE
         DO 100 L = ZROREG, NUMREG
            IF (S .GT. SBND(L)) THEN
               TEMP = LNA(L) + LPHIM(L) + MM(L) * ( - ALOG(S)
     &                  +  ALOG (KRATIO) + LNZ)
               IF (TEMP .GT. 86.0) THEN
                  TEMP = 86.0
               ENDIF
               GETLIF = EXP (TEMP)
               GOTO 150
            ENDIF
 100     CONTINUE
      ENDIF
 150  CONTINUE

      GTLIFE = GETLIF

      RETURN
      END


C**************************************************************************


C  SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C  M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB88
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE SORTM (ALLM, NUMREG, NUM)

C   INPUTS:  ALLM, NUMREG, NUM
C  OUTPUTS:  ALLM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER I, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG

      PARAMETER (MAXMM = 20001, MAXREG = 3)

      LOGICAL INORDR

      REAL    ALLM(MAXMM, MAXREG), TEMP

C                  LIST OF VARIABLES
C
C  ALLM()     2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
C  I          CONTROLS INSERTION POINTER
C  INC        SORT INCREMENT VARIABLE
C  INORDR     FLAG TO INDICATE WHETHER SORT IS FINISHED
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXMM      MAXIMUM NUMBER OF M'S TO BE SORTED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
```

```
C  NUM        NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  TEMP       TEMPORARY SORTING VARIABLE


      DO 400 L = 1, NUMREG

  5      INC = NUM
 10      IF (INC .GT. 1) THEN
            INC = INC / 2
 20         INORDR = .TRUE.

            DO 300 I = 1, (NUM - INC)
               IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN
                  TEMP = ALLM(I,L)
                  ALLM(I,L) = ALLM(I + INC, L)
                  ALLM(I + INC, L) = TEMP
                  INORDR = .FALSE.
               ENDIF
300         CONTINUE

            IF (.NOT. INORDR) GOTO 20
            GOTO 10
         ENDIF

400   CONTINUE

      RETURN
      END

C***************************************************************************

C************************************************************************
C
C  FUNCTION RAINF3 CALCULATES THE TIME (in missions) TO FAILURE FOR
C  THE GIVEN STRAIN-TIME HISTORY
C
C  PROGRAMMER:  L. SHIRAISHI, L. NEWLIN
C        DATE:  27MAR90
C     VERSION:  1.1 (BLDLCF V3.1, V3.2, V3.3, V3.4  MATCHR V8.4, V8.5)
C
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C************************************************************************

      FUNCTION RAINF3 (SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM,
     &                 KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO)

C  INPUTS:    SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO,
C             LNZ, SBND, ZROREG, NUMREG, SZERO
C  OUTPUTS:   RAINF3

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER  MAXREG, MAXM

      PARAMETER (MAXREG = 3, MAXM = 50)

      INTEGER  I, INDEX(MAXM), IOUT, J, JMAX, K, M, N, NEWTOT, NUMREG,
     &         ZROREG

      REAL     CHKFT, E(MAXM), GTLIFE, INVLIF(MAXM), KRATIO,
     &         LIFE(MAXM), LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
     &         MM(0:MAXREG), PERIOD, RAINF3, S(MAXM), SBND(0:MAXREG),
     &         SEFF(MAXM), SEFFM(2, MAXM), SEFMAX, SP(MAXM),
     &         SRANGE(MAXM), SUMDAM, SZERO, TEST1(MAXM), TEST2(MAXM),
     &         TRUNC, WEXP
```

```
C                        LIST OF VARIABLES
C
C  RAINF3       TIME TO FAILURE FOR THE GIVEN TIME HISTORY
C
C input variables:
C
C  SEFF(M)      EFFECTIVE STRAINS BEFORE FILTERING/RAINFLOW
C  M            TOTAL NUMBER OF STRAIN DATA POINTS PER PERIOD
C  TRUNC        VALUE USED TO FILTER OUT NOISE
C  PERIOD       TIME IN SECONDS FOR ONE PERIOD
C  WEXP         WALKER EXPONENT
C
C
C intermediate variables:
C
C  MAXM         MAXIMUM NUMBER OF POINTS ALLOWED IN STRAIN-TIME HISTORY
C                 ARRAYS
C  SEFMAX       LARGEST EFFECTIVE STRAIN
C  JMAX         INDEX (LOCATION) OF SEFMAX IN SEFF()
C  I,J,K        COUNTERS FOR VARIOUS DO LOOPS
C  SP(M+1)      RESEQUENCED EFFECTIVE STRAINS; # OF PTS = M+1
C  INDEX(MAXM), TEST1(MAXM), TEST2(MAXM)
C                 INTERMEDIATE CALCULATION ARRAYS USED DURING FILTERING
C  S(NEWTOT)    FILTERED EFFECTIVE STRAINS
C  NEWTOT       TOTAL NUMBER OF EFFECTIVE STRAIN VALUES AFTER FILTERING
C  E()          HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
C  N            NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
C  SEFFM(2,N)   EFFECTIVE STRAINS AFTER RESEQUENCING/FILTERING/RAINFLOW
C                    SEFFM(1,I) = sigma max,eff,i
C                    SEFFM(2,I) = sigma min,eff,i
C  SRANGE(N)    SRANGE(I) = EQUIVALENT STRAIN RANGE FOR CYCLE I
C  GTLIFE       REAL FUNCTION THAT CALCULATES FATIGUE LIFE FOR A GIVEN STRAIN
C  LIFE(N)      LIFE(I) = CALCULATED LIFE FOR STRAIN LEVEL SRANGE(I)
C  INVLIF(N)    INVLIF(I) = 1/LIFE(I); DAMAGE FRACTION
C  SUMDAM       SUM OF ALL THE DAMAGE FRACTIONS
C  CHKFT        DUMMY VARIABLE USED TO PRINT OUT RAINF3 RESULT
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  LNA()        1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C  LNZ          NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  LPHIM()      1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
C                 PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()         1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  SBND()       1-D ARRAY CONTAINING THE STRAIN VALUES (%, R = - 1.0)
C                 CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C                 REGION CONTAINED IN NBND() CORRECTED BY PHI, KRATIO,
C                 AND LNZ
C  SZERO        STRAIN TENSILE POINT, So (%)
C  ZROREG       Zero Region — VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                 REGION

C  dump input data
      if (iout.eq.20) then
        write(8,*)' rainf3 inputs'
        write(8,*)'m      :',m,'      period:',period

        write(8,*)'wexp :', wexp
        write(8,*)'numreg     :',numreg,'zroreg :',zroreg
        write(8,*)'szero  :',szero,'kratio  :',kratio,'lnz  :',lnz
        write(8,*)'lna(i),     mm(i),      lphim(i),      sbnd(i)'
        write(8,*) (lna(i), mm(i), lphim(i), sbnd(i), i=zroreg,numreg)
        write(8,*)' '
      endif

C  INITIALIZE ARRAYS

      DO 50 I = 1, MAXM
         SP(I) = 0.0
         S(I) = 0.0
```

```
         E(I) = 0.0
         SEFFM(1,I) = 0.0
         SEFFM(2,I) = 0.0
         SRANGE(I) = 0.0
         LIFE(I) = 0.0
         INVLIF(I) = 0.0
         INDEX(I) = 0
         TEST1(I) = 0.0
         TEST2(I) = 0.0
   50 CONTINUE

C********************* B E G I N    R E S E Q U E N C E ************************
C  RESEQUENCE effective strains (needed for rainflow analysis);
C   largest effective strain is placed at beginning and end of SP(M+1)

C   find SEFMAX, the largest sigma,eff, and JMAX, its location within SEFF(M)
      SEFMAX = -1.0E+20
      DO 200 I=1,M
         IF ( SEFF(I) .GT. SEFMAX ) THEN
            SEFMAX = SEFF(I)
            JMAX = I
         ENDIF
 200  CONTINUE

C   assign all points from JMAX out, to the beginning of SP()
      DO 210 I = 1, M-JMAX+1
         J = JMAX-1 + I
         SP(I) = SEFF(J)
 210  CONTINUE

C   assign points before JMAX to the end of SP()
      J = 0
      DO 220 I = M-JMAX+2, M
         J = J + 1
         SP(I) = SEFF(J)
220   CONTINUE
      SP(M+1) = SEFF(JMAX)
      if (iout.eq.20) then
         write(8,*)'sefmax:',sefmax,'     jmax:',jmax
         write(8,*)'sp(m+1):',(sp(i),i=1,m+1)
      endif


C********************** E N D    R E S Q U E N C E ****************************
C*********************** B E G I N    F I L T E R ******************************
C  FILTER the resequenced effective strains, leaving only peaks and valleys
C   (excursions larger than TRUNC are deleted during rainflow counting) in
C   S(NEWTOT), where NEWTOT is the new number of points
C

      DO 300 I = 2, M
         TEST1(I) = SP(I-1) - SP(I)
         TEST2(I) = TEST1(I) * (SP(I) - SP(I+1))
  300 CONTINUE

C       if (iout .eq. 20) then
C          do 305 i = 2, m
C             write(8,*) 'test1 = ', test1(i), ' test2 = ', test2(i)
C  305     continue
C       endif

      K = 1
      INDEX(1) = 1

      DO 310 I = 2, M
         IF ((TEST1(I) .NE. 0) .AND. (TEST2(I) .LT. 0)) THEN
            K = K + 1
            INDEX(K) = I
         ENDIF
  310 CONTINUE

      NEWTOT = K + 1
      INDEX(NEWTOT) = M + 1
```

```
      DO 320 I = 1, NEWTOT
         K = INDEX(I)
         S(I) = SP(K)
  320 CONTINUE

      if (iout.eq.20) then
         write(8,*)'newtot:',newtot
         write(8,*)'s(newtot):',(s(i),i=1,newtot)
      endif


C*************************** E N D    F I L T E R ********************************
C********************** B E G I N    R A I N F L O W ****************************
C RAINFLOW ANALYSIS to identify cycles within effective strain data, S(NEWTOT);
C   places each cycle's max and min values into SEFFM(2,N)
C
C   counters: I counts # of cycles found, J counts how many S()'s counted,
C             K accumulates unmatched points
      I = 0
      J = 0
      K = 0
 400  CONTINUE
      J = J+1
      K = K+1
C   check J to avoid reading beyond end of filtered strain data
      IF ( J .GT. NEWTOT ) GOTO 499

C   read strain point into a holding array to be checked for cycles
      E(K) = S(J)
 410  IF ( K .LT. 3 ) GOTO 400
      IF ( ABS( E(K) - E(K-1) ) .LT. ABS( E(K-1) - E(K-2) ) ) GOTO 400
C    if not, then a cycle has been found, but we need to check for truncation
      IF (ABS (E(K-1) - E(K-2)) .GT. TRUNC) THEN
C        cycle is large enough to save
         I = I+1
         SEFFM(1,I) = AMAX1( E(K-1), E(K-2) )
         SEFFM(2,I) = AMIN1( E(K-1), E(K-2) )
      ENDIF
C   discard points K-1 and K-2, and decrement the counter of unmatched points
      E(K-2) = E(K)
      K = K-2
C   return for more counting
      GOTO 410

  499 CONTINUE
C   N equals the final number of cycles found
      N = I

      if (iout.eq.20) then
         write(8,*)'N :',n
         write(8,*)'seffm(2,n):'
         do 12 i=1,n
            write(8,*) seffm(1,i), seffm(2,i)
 12      continue
      endif

      IF (N .EQ. 0) THEN
C        truncation filter value too large — no cycles left
         SUMDAM = 1.0E-36
         GOTO 710
      ENDIF

C
C********************** E N D    R A I N F L O W ***************************


C  calculate equivalent strain range
C
      DO 500 I=1,N
         SRANGE(I) = (SEFFM(1,I) - SEFFM(2,I))
     &               * ((SEFFM(1,I) - SEFFM(2,I)) / (2.0 * SEFFM(1,I)))
     &               ** (WEXP - 1.0)
 500  CONTINUE
      if (iout.eq.20) write(8,*)'srange(n) :',(srange(i),i=1,n)
      if (iout.eq.25) write(8,*) (srange(i),i=1,n), ' ',
```

```
     &    exp(lphim(1)/mm(1))

C  calculate lives and damage fractions: LIFE(N) and INVLIF(N)
C
      DO 600 I=1,N
         LIFE(I) = GTLIFE (SRANGE(I), MM, LNA, LPHIM, KRATIO, LNZ,
     &                      SBND, ZROREG, NUMREG, SZERO)
 600  CONTINUE
      DO 650 I=1,N
         INVLIF(I) = 1.0 / LIFE(I)
 650  CONTINUE
      if (iout.eq.20) then
         do 14 i=1,n
            write(8,*)'life(n):',life(i),'      invlif(n):',invlif(i)
 14      continue
      endif

C  Miner's Rule - sum the damage fractions
C
      SUMDAM = 0.0
      DO 700 I=1,N
         SUMDAM = SUMDAM + INVLIF(I)
  700 CONTINUE
  710 CONTINUE
      if (iout.eq.20) write(8,*)'sumdam:',sumdam

C  calculate fatigue life (time to failure)
C
      RAINF3 = PERIOD / SUMDAM

      if (iout.eq.15) then
         chkft=period/sumdam
         write(8,*)' rainf3 life',chkft
         write(8,*)
      endif

      RETURN
      END
```

7.2.5 Program BLDLCF V3.4B1.3 Listing

BLDLCF Version 3.4B1.3

Program BLDLCF V3.4B1.3 Listing Temporal Order

```
C***********************************************************************
C  PROGRAM BLDLCF CONTROLS THE FLOW OF LOGIC OF THE LOW CYCLE
C  FATIGUE ANALYSIS OF THE TURBINE BLADE FOIL PROBLEM
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  5FEB92      COMMENTS:  17APR92
C     VERSION:  3.4B1.3 (MATCHR VB1.3, RAINF3 V1.1, INSORT V2.1)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      PROGRAM BLDLCF

C  SUBPROGRAMS:  INFAGG, PEB, PRYRV, BETAGN, NORMGN, WORSTN,
C                TRMNAT, BLDLIF, INSORT, SORTM, EXPTCD
C        FILES:  1:BLDLCD-OLD; 3:BLDLCO-NEW; 5:RELATD-OLD; 6:RELATO-NEW;
C                7:DUMP-NEW;   8:IOUTPR-NEW; 9:LOWLIF-NEW;
C                NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

C     IMPLICIT NONE

      INTEGER  MAXBLF, MAXDAT, MAXLIF, MAXM, MAXMM, MAXREG

      PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLIF = 10000,
     &           MAXM = 50, MAXMM = 20001, MAXREG = 3)

      COMMON   IOUT

      INTEGER  BLFPOS(MAXBLF), I, IOUT, J, K, L, MCOUNT, MID,
     &         MPROC, NBLIFE, NHYPER, NLIFE, NLIFET, NMED,
     &         NPTS(MAXREG), NSYM, NTIME, NUMREG, TSUBI, VARY,
     &         ZROREG

      DOUBLE PRECISION  RAND

      REAL     ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGK1, BLDLIF,
     &         BLFPER(MAXBLF), BZERO, DUM, EBEND, EBENDA, EBENDB,
     &         EMNOM, EPSL, EPSW, ETHNOM(MAXM), FAA, FAB, FAC, FACTR,
     &         FAD, FAE, FAF, FAERRM, FAERRS, FD1A, FD1B, FD1C, FD1D,
     &         FD1E, FD1F, FD2A, FD2B, FD3A, FD3B, FDERRM, FDERRS,
     &         FIFTY, FTU, FTY, HGAS, HGASA, HGASB, HGASR, HGASR1,
     &         HGASR2, HGAST, HGAST1, HGAST2, KRATIO, LAMA, LAMAA,
     &         LAMAB, LAMDA, LAMDAA, LAMDAB, LAMG, LAMGA, LAMGB, LAMGR,
     &         LAMGR1, LAMGR2, LAMGT, LAMGT1, LAMGT2, LAMP, LAMPA,
     &         LAMPB, LAMTM, LAMTMA, LAMTMB, LIFEL(MAXLIF),
     &         LIFEW(MAXLIF), LIFL, LIFW, LNA(0:MAXREG), LNZ
      REAL     LPHIM(0:MAXREG), MANAL, MANALA, MANALB, MEDKB(0:MAXREG),
     &         MEDM(MAXREG), MEDMB(0:MAXREG), MM(0:MAXREG), MODER1,
     &         MODER2, MU(MAXREG), NBND(0:MAXREG), NEWLIF,
     &         NF(MAXDAT, MAXREG), NOMSPD, PERIOD, PHI, PSIG, PVAR,
     &         RANGEM(2, MAXREG), RESID(MAXDAT), RPM(MAXM),
     &         SBND(0:MAXREG), SIG(MAXREG), SLOPE, SLOPEA, SLOPEB,
     &         SLOPR, SLOPR1, SLOPR2, SLOPT, SLOPT1, SLOPT2, SPEED,
     &         SPEEDM, SPEEDS, STR(MAXDAT, MAXREG), SZERO, TANAL,
     &         TANALA, TANALB, TGAS, TGASA, TGASB, TGASR, TGASR1,
     &         TGASR2, TGAST, TGAST1, TGAST2, TRBIGK(0:MAXREG),
     &         TRSBND(0:MAXREG), TRUNC, TSTART, TSTMU, TSTSIG, WEXP, Z

C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

      OPEN (1,  FILE = 'BLDLCD', STATUS = 'OLD')
      OPEN (3,  FILE = 'BLDLCO', STATUS = 'NEW')
      OPEN (7,  FILE = 'DUMP',   STATUS = 'NEW')
      OPEN (8,  FILE = 'IOUTPR', STATUS = 'NEW')
      OPEN (9,  FILE = 'LOWLIF', STATUS = 'NEW')


      READ(1,*) RAND
      WRITE(8,*) '                          RANDOM NUMBER SEED =', RAND
      READ(1,*) IOUT
      WRITE(8,*) 'IOUT (MATCHR = 10, BLDLCF = 15, RAINF3 = 20)=', IOUT
      READ(1,*) NLIFE
      WRITE(8,*) '                             INNER LOOP SIZE =', NLIFE
      READ(1,*) NHYPER
```

```
      WRITE(8,*) '                                    OUTER LOOP SIZE =',NHYPER
      READ(1,*) NSYM
      WRITE(8,*) '                                    SYMMETRY NUMBER =', NSYM
      READ(1,*) VARY
      WRITE(8,*) '                       TYPE OF S/N VARIATION DESIRED '
      WRITE(8,*) '                                    (4 - BOOTSTRAP) =',VARY
      READ(1,*) NMED
      WRITE(8,*) '                 NORMAL MEDIAN CURVE (0 - NO, 1 - YES) =',NMED
      READ(1,*) MPROC
      WRITE(8,*) '                  MATERIALS PROCESS VARIATION DESIRED'
      WRITE(8,*) '                                  (0 - NO, 1 - YES) =', MPROC

      IF (VARY .NE. 4) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF S/N VARIATION DESIRED'
         CALL TRMNAT
      ENDIF
      IF ((NMED .NE. 0) .AND. (NMED .NE. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID RESPONSE TO NORMAL MEDIAN ',
     &              'CURVE QUESTION'
         CALL TRMNAT
      ENDIF

      IF ((MPROC .LT. 0) .OR. (MPROC .GT. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF MATERIALS PROCESS ',
     &              'VARIATION DESIRED'
         CALL TRMNAT
      ENDIF

      READ(1,*) NBLIFE
      IF (NBLIFE .GT. 0) READ(1,*) (BLFPER(J), J = 1, NBLIFE)

C ** READ DATA FROM BLDLCD

      READ(1,*) HGASA,  HGASB,  HGASR1, HGASR2, HGAST1, HGAST2,
     &          TGASA,  TGASB,  TGASR1, TGASR2, TGAST1, TGAST2,
     &          SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
     &          LAMGA,  LAMGB,  LAMGR1, LAMGR2, LAMGT1, LAMGT2,
     &          TSUBI,  SPEEDM, SPEEDS,
     &          FAERRM, FAERRS, TSTMU,  TSTSIG,
     &          FDERRM, FDERRS,
     &          EBENDA, EBENDB, LAMPA,  LAMPB,
     &          MANALA, MANALB, LAMAA,  LAMAB,
     &          TANALA, TANALB, LAMDAA, LAMDAB,
     &          LAMTMA, LAMTMB
      READ(1,*) EMNOM,  NOMSPD, PERIOD, TRUNC, NTIME, WEXP
      READ(1,*) FAA, FAB, FAC, FAD, FAE, FAF,
     &          FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
     &          FD2A, FD2B,
     &          FD3A, FD3B

      IF (NTIME .GT. MAXM) THEN
         WRITE(8,*) 'ERROR:  STRAIN-TIME HISTORY TOO LARGE'
         CALL TRMNAT
      ENDIF

      DO 20 I = 1, (NTIME - 1)
         READ(1,*) RPM(I), ETHNOM(I)
   20 CONTINUE

C ** ECHO DATA TO BLDLCO

      WRITE(3,900)
      WRITE(3,901) HGASA,  HGASB,  HGASR1, HGASR2, HGAST1, HGAST2,
     &             TGASA,  TGASB,  TGASR1, TGASR2, TGAST1, TGAST2
      WRITE(3,902) SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
     &             LAMGA,  LAMGB,  LAMGR1, LAMGR2, LAMGT1, LAMGT2
      WRITE(3,903) TSUBI,  SPEEDM, SPEEDS, FAERRM, FAERRS,
     &             TSTMU,  TSTSIG, FDERRM, FDERRS
      WRITE(3,904) EBENDA, EBENDB, LAMPA,  LAMPB,  MANALA, MANALB,
     &             LAMAA,  LAMAB,  TANALA, TANALB
      WRITE(3,905) EXP(LAMDAA), EXP(LAMDAB), EXP(LAMTMA), EXP(LAMTMB)
      WRITE(3,906) EMNOM,  NOMSPD, PERIOD, TRUNC, NTIME, WEXP
      WRITE(3,907) FAA, FAB, FAC, FAD, FAE, FAF,
     &             FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
     &             FD2A, FD2B,
```

```
     &              FD3A, FD3B

      DO 25 I = 1, (NTIME - 1)
         WRITE(3,908) RPM(I), ETHNOM(I)
   25 CONTINUE

C ** CALL INFAGG TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECT
C       OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS

      CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, ZROREG, NUMREG,
     &             NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR,
     &             MEDMB, MEDKB, RESID)

      IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)

      MCOUNT = 0

C ** INITIALIZE VARIABLES

      DO 35 K = 1, MAXLIF
         LIFEW(K) = 1.0E+36
         LIFEL(K) = 1.0E+36
   35 CONTINUE

      NLIFET = NHYPER * NLIFE

C ** OUTER LOOP — THIS LOOP SAMPLES HYPER-PARAMETER SETS

      DO 150 K = 1, NHYPER

C ** CALL PRYRV TO OBTAIN RHO,THETA PAIRS FOR INNER LOOP CALCULATIONS

         CALL PRYRV (RAND, HGASR1, HGASR2, HGAST1, HGAST2, HGASR, HGAST)
         CALL PRYRV (RAND, TGASR1, TGASR2, TGAST1, TGAST2, TGASR, TGAST)
         CALL PRYRV (RAND, SLOPR1, SLOPR2, SLOPT1, SLOPT2, SLOPR, SLOPT)
         CALL PRYRV (RAND, LAMGR1, LAMGR2, LAMGT1, LAMGT2, LAMGR, LAMGT)
         CALL PRYRV (RAND, MANALA, MANALB, TANALA, TANALB, MANAL, TANAL)

C ** CALL PEB TO PERFORM THE BOOTSTRAPPING ASPECT OF THE
C       MATERIALS CHARACTERIZATION MODEL CALCULATIONS

         CALL PEB (NPTS, NUMREG, ZROREG, RAND, NBND, STR, MEDMB,
     &             MEDKB, RESID, BIGK, BZERO, MM, SBND)

C ** OBTAIN MATERIALS PROCESS VARIATION PARAMETERS IF DESIRED

         CALL NORMGN (RAND, 0.0, PSIG, LNZ)

         IF (MPROC .EQ. 1) THEN
            Z = EXP (LNZ)
         ELSE
            KRATIO = 1.0
            Z = 1.0
            LNZ = 0.0
         ENDIF

         MCOUNT = MCOUNT + 1
         DO 175 L = 1, NUMREG
            ALLM(MCOUNT, L) = MM(L)
  175    CONTINUE

C ** INNER LOOP — THIS LOOP GENERATES BLADE FAILURE TIMES

         DO 200 I = 1, NLIFE

C ** INITILIZE S/N CURVE PARAMETERS

            DO 225 L = 0, MAXREG
               LNA(L) = 0.0
               LPHIM(L) = 0.0
               TRSBND(L) = 0.0
  225       CONTINUE

C ** SELECT DRIVERS FOR CALCULATING LIFE
```

```
            CALL BETAGN (RAND, HGASR, HGAST, HGASA,  HGASB,  HGAS)
            CALL BETAGN (RAND, TGASR, TGAST, TGASA,  TGASB,  TGAS)
            CALL BETAGN (RAND, SLOPR, SLOPT, SLOPEA, SLOPEB, SLOPE)
            CALL BETAGN (RAND, LAMGR, LAMGT, LAMGA,  LAMGB,  LAMG)

            CALL NORMGN (RAND, SPEEDM, SPEEDS, SPEED)
            CALL NORMGN (RAND, FAERRM, FAERRS, MODER1)
            CALL NORMGN (RAND, TSTMU,  TSTSIG, TSTART)
            CALL NORMGN (RAND, FDERRM, FDERRS, MODER2)

            CALL PRYRV (RAND, EBENDA,EBENDB,LAMPA, LAMPB, EBEND, LAMP)
            CALL PRYRV (RAND, LAMAA, LAMAB, LAMAA, LAMAB, LAMA,  DUM)
            CALL PRYRV (RAND, LAMDAA,LAMDAB,LAMTMA,LAMTMB,LAMDA, LAMTM)
            LAMDA = EXP (LAMDA)
            LAMTM = EXP (LAMTM)

            CALL WORSTN (RAND, NSYM, BZERO, MM, EPSW, EPSL)

            IF ((VARY .EQ. 0) .OR. (VARY .EQ. 4)) PHI = 1.0

            IF (IOUT .EQ. 15) THEN
               WRITE(8,*) 'HGAS =', HGAS, ' TGAS =', TGAS
               WRITE(8,*) 'SLOPE =', SLOPE, ' LAMG = ', LAMG
               WRITE(8,*) 'LAMP =',LAMP,' EBEND =',EBEND,' LAMA =',LAMA
               WRITE(8,*) 'SPEED = ', SPEED,' LAMDA =',LAMDA
               WRITE(8,*) 'LAMTM =',LAMTM,' PHI =',PHI
               WRITE(8,*) 'MANAL =',MANAL,' TANAL =',TANAL
               WRITE(8,*) 'TSTART =',TSTART,' MODER1 =',MODER1,
     &                    ' MODER2 =',MODER2
               WRITE(8,*) 'EPSW = ', EPSW, ' EPSL = ', EPSL
            ENDIF

C ** CALCULATE REGION DEPENDENT S/N CURVE PARAMETERS

            FACTR = PHI * KRATIO * Z

            DO 235 L = ZROREG, NUMREG
              TRSBND(L) = FACTR * SBND(L)
              TRBIGK(L) = BIGK(L)
  235       CONTINUE
            TRSBND(0) = SBND(0)

            IF (ZROREG .EQ. 0) CALL KOMO (SZERO, BIGK, MM, NBND,
     &          TRSBND, TRBIGK, FACTR, NUMREG)

            DO 250 L = ZROREG, NUMREG
               LNA(L) = MM(L) * ALOG(TRBIGK(L))
               LPHIM(L) = MM(L) * ALOG(PHI)
C                TRSBND(L) = SBND(L) * PHI * KRATIO * Z
               IF (IOUT .EQ. 15) THEN
                  WRITE(8,*) 'L =',L,' MM =',MM(L),' BIGK =',TRBIGK(L)
                  WRITE(8,*) 'LNA =', LNA(L), ' PHI =', PHI
                  WRITE(8,*) 'LPHIM =', LPHIM(L), ' SBND =', SBND(L)
                  WRITE(8,*) 'KRATIO = ', KRATIO, ' Z = ', Z
                  WRITE(8,*) 'TRSBND =', TRSBND(L), ' FACTR = ', FACTR
               ENDIF
  250       CONTINUE

C ** CALL BLDLIF TO OBTAIN BLADE LCF LIFE

            NEWLIF = LAMDA * LAMTM * BLDLIF (TGAS, HGAS, FAA, FAB, FAC,
     &               FAD, FAE, FAF, MODER1, RPM, TSUBI, SPEED, SLOPE,
     &               TSTART, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
     &               MODER2, FD2A, FD2B, FD3A, FD3B, ETHNOM, MANAL,
     &               LAMP, NOMSPD, EMNOM, TANAL, LAMA, LAMG, EBEND,
     &               NTIME, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM,
     &               KRATIO, LNZ, TRSBND, ZROREG, NUMREG, SZERO)

            LIFW = EPSW * NEWLIF
            LIFL = EPSL * NEWLIF

            IF (IOUT .EQ. 15) THEN
               WRITE(8,*) 'NEWLIF = ', NEWLIF
               WRITE(8,*) 'LIFW = ', LIFW, ' LIFL = ', LIFL
            ENDIF
```

```
            IF (NLIFET .GE. 100) THEN
               CALL INSORT (LIFW, LIFEW, NLIFET)
               CALL INSORT (LIFL, LIFEL, NLIFET)
            ENDIF

  200    CONTINUE

  150 CONTINUE

      IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES TO FILE LOWLIF

         DO 300 J = 1, (NLIFET / 100)
            WRITE(9,*) J, FLOAT(J)/FLOAT(NLIFET), LIFEW(J), LIFEL(J)
  300    CONTINUE

C ** INITIALIZE VARIABLE BLFPOS()

         DO 325 J = 1, MAXBLF
            BLFPOS(J) = 0
  325    CONTINUE

         FIFTY = 0.50E0

C ** PRINT EMPIRICAL BLIVES

         WRITE(3,925)

         DO 350 J = 1, NBLIFE
            BLFPOS(J) = NINT (BLFPER(J) * FLOAT (NLIFET))
            WRITE(3,926) BLFPER(J), LIFEW(BLFPOS(J)), LIFEL(BLFPOS(J))
  350    CONTINUE
         WRITE(3,926) FIFTY, LIFEW(NLIFET/2), LIFEL(NLIFET/2)

      ENDIF

C ** CALCULATE NORMAL MEDIAN CURVE IF DESIRED

      IF (((VARY .EQ. 3) .AND. (NMED .EQ. 1)) .OR. (VARY .EQ. 4)) THEN

         CALL SORTM (ALLM, NUMREG, MCOUNT)

         MID = MCOUNT / 2
         DO 400 L = 1, NUMREG
            MEDM(L) = ALLM(MID,L)
  400    CONTINUE

         CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG,
     &                NBND, BIGK1, BZERO)

      ENDIF


C ** FORMAT STATEMENTS TO ECHO INPUT DATA TO BLDLCO

  900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &       'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &       'NASA Contract NAS7-918 is acknowledged.',////,
     &       33X,'INPUT DATA',
     &       ///,14X,'DRIVERS',25X,'PARAMETER DISTRIBUTIONS',
     &       //,48X,'RHO',16X,'THETA')

  901 FORMAT(/,2X,'Hgas',13X,'Be(',F5.0,',',F6.0,')',5X,
     &       'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')',
     &       //,2X,'Tgas (deg R)',5X,'Be(',F5.0,',',F6.0,')',5X,
     &       'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')

  902 FORMAT(/,2X,'DECEL SLOPE',6X,'Be(',F5.0,',',F6.0,')',5X,
     &       'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')',
     &       //,2X,'Tgas UNCERT.',5X,'Be(',F5.2,',',F6.2,')',5X,
     &       'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')
```

```
903 FORMAT(//,50X,'N( MEAN, STD. DEV.)',
     &      //,2X,'ROTOR SPEED VARIATION (rpm) AT TIME T',I1,
     &      10X,'N(',F8.1 ,',',F7.1,')',//,
     &      2X,'Faccel MODELING ERROR',27X,'N(',F4.1,',',E11.4,')',
     &      //,2X,'STARTING DECEL TEMPERATURE (deg R)',14X,
     &      'N(',F8.2,',',F7.2,')',//,
     &      2X,'Fdecel MODELING ERROR',27X,'N(',F4.1,',',E11.4,')')

904 FORMAT(//,2X,'STRAIN DUE TO GAS BENDING (%)',17X,
     &      'U(',F8.5,',',F9.5,')',
     &      //,2X,'LAMBDA BLADE PULL',29X,
     &      'U(',F8.5,',',F9.5,')',
     &      //,2X,'MECHANICAL ANALYSIS FACTOR',20X,
     &      'U(',F8.5,',',F9.5,')',
     &      //,2X,'COEFFICIENT OF THERMAL EXPANSION FACTOR',7X,
     &      'U(',F8.5,',',F9.5,')',
     &      //,2X,'THERMAL ANALYSIS FACTOR',23X,
     &      'U(',F8.5,',',F9.5,')')

905 FORMAT(//,2X,'DAMAGE MODEL ACCURACY',23X,
     &      'U(ln',F8.5,', ln',F8.5,')',
     &      //,2X,'TMF MODEL ACCURACY',26X,
     &      'U(ln',F8.5,', ln',F8.5,')')

906 FORMAT(////,20X,'OTHER STRAIN HISTORY INPUT',
     &      ///,2X,'NOMINAL MECHANICAL STRAIN (%)',23X,F6.4,
     &      //,2X,'NOMINAL ROTOR SPEED (rpm)',23X,F6.0,
     &      //,2X,'STRAIN-TIME HISTORY PERIOD (missions)',14X,F5.2,
     &      //,2X,'STRAIN-TIME HISTORY NOISE FILTER (%)',16X,F7.5,
     &      //,2X,'NUMBER OF POINTS IN HISTORIES',19X,I5,
     &      //,2X,'WALKER EXPONENT',36X,F5.2)

907 FORMAT(////,6X,'COEFFICIENTS OF ACCELERATION AND DECELERATION ',
     &      'FUNCTIONS',//,2X,'THERMAL STRAIN AT STARTUP (%):',/,5X,
     &      'Faccel(Tgas, Hgas) = ',E13.6,' + ',E13.6,' * Tgas + ',
     &      /,15X,E13.6,' * Hgas + ',E13.6,' * Tgas ** 2 + ',
     &      /,15X,E13.6,' * Hgas**2 + ',E13.6,' * Tgas * Hgas',
     &      //,2X,'THERMAL STRAIN AT SHUTDOWN (%):',/,5X,
     &      'Fdecel1(m, Tstart) = ',E13.6,' + ',E13.6,' * Tstart + ',
     &      /,15X,E13.6,' * m + ',E13.6,' * Tstart ** 2 + ',
     &      /,15X,E13.6,' * m ** 2 + ',E13.6,' * Tstart * m',
     &      //,2X,'TIME AT SHUTDOWN (sec):',
     &      /,5X,'Fdecel2(m, Tstart) = ',E13.6,' + ','(Tstart - ',
     &      E13.6,' ) / m',
     &      //,2X,'ROTOR SPEED AT SHUTDOWN (rpm):',
     &      /,5X,'Fdecel3(t) = ',E13.6,' + ',E13.6,' * t',
     &      /////,20X,'STRAIN HISTORY INFORMATION',
     &      ///,5X,'ROTOR SPEED',5X,'THERMAL STRAIN',
     &      /,9X,'rpm',15X,'(%)',/)

908 FORMAT(7X,F7.1,9X,F9.6)

925 FORMAT(///,2X,'B LIVES:          WEIBULL          LOGNORMAL',/)

926 FORMAT(2X,F7.5,5X,E13.6,5X,E13.6)


    STOP
    END


C*********************************************************************
C                  SAMPLE 'BLDLCD' INPUT FILE
C*********************************************************************
C 675.............................RANDOM NUMBER SEED
C 0...............................OUTPUT DUMP CONTROLLER
C 100.............................INNER LOOP SIZE
C 200.............................OUTER LOOP SIZE
C 50..............................SYMMETRY NUMBER
C 4...............................BOOTSTRAP S/N VARIATION
C 0...............................NORMAL MEDIAN NOT REQUIRED
C 0...............................MAT. PROC. VAR. NOT REQUIRED
C 3...............................NUMBER OF BLIVES REQUESTED
```

```
C 0.0001.............................B.01 LIFE
C 0.001..............................B.1 LIFE
C 0.01...............................B1 LIFE
C  676. 2730. 0.5  0.5  0.0  0.0....Hgas            (A,B) (R1,R2) (T1,T2)
C  800. 2000. 0.5  0.5  0.0  0.0....Tgas            (A,B) (R1,R2) (T1,T2)
C 2730. 2730. 0.5  0.5  0.0  0.0....DECEL SLOPE     (A,B) (R1,R2) (T1,T2)
C 0.80  1.20  0.5  0.5  0.0  0.0....Tgas UNCERTAINTY FACTOR
C 5   37592.   507..................ROTOR SPEED VARIATION PARAMETERS:
C                                     i, MEAN, STD.DEV. (NORMAL DIST.)
C 0.0     0.020......................Faccel MODELING ERROR MEAN & STD.DEV.
C 1640.0    40.67....................DECEL Tstart MEAN & STANDARD DEVIATION
C 975.3     28.6.....................STANDARD RESPONSE PROBE MEAN & STD DEV
C 0.0    0.003.......................Fdecel MODELING ERROR MEAN & STD DEV
C 0.0       0.0......................STRAIN DUE TO GAS BENDING (%)
C 0.96      1.04.....................LAMBDA BLADE PULL
C 0.80      1.20.....................MECHANICAL ANALYSIS ACCURACY FACTOR
C 0.975     1.025....................COEFFICIENT OF THERMAL EXPANSION
C 0.70      1.30.....................THERMAL ANALYSIS ACCURACY FACTOR
C -0.693147   0.563283...............DAMAGE ACCUMULATION MODEL ACCURACY
C  0.00       0.00...................TMF MODEL ACCURACY
C 0.295    38482.....................NOMINAL MECH. STRAIN & ROTOR SPEED (%,RPM)
C 1.0................................STRAIN-TIME HISTORY PERIOD (MISSIONS)
C 0.000..............................STRAIN-TIME HISTORY NOISE FILTER (%)
C 6..................................NUMBER OF POINTS IN STRAIN-TIME HISTORY
C 0.5................................WALKER EXPONENT
C
C COEFFICIENTS FOR STARTUP RESPONSE SURFACE FOR THERMAL STRAIN:
C    Faccel(Tgas,Hgas) = A + B * T + C * H + D * T**2 + E * H**2 + F * T * H
C       A          B           C            D            E            F
C 0.00727362 0.000067442 -0.000059109 -3.52929E-08 1.07611E-08 -2.74419E-08
C
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR THERMAL STRAIN:
C    Fdecel1(m,Tstart) = A + B * Tstart + C * m + D * Tstart ** 2
C                      + E * m ** 2 + F * Tstart * m
C       A            B             C           D       E          F
C -0.132623    0.000227427   -0.000059290    0.00    0.00    4.71714E-08
C
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR TIME:
C    Fdecel2(m,Tstart) = A + (Tstart - B) / m
C       A            B
C   0.20          950.0
C
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR RPM:
C    Fdecel3(t) = A + B * t
C       A             B
C 30523.07       -21846.15
C
C RPM(TIME)   THERMAL STRAIN (%).....STRAIN HISTORY INFORMATION
C   225.8          0.0
C  3025.1         -0.196921
C  6138.8          0.146025
C  8309.0         -0.200128
C     0.0          0.007393
C
C 'RT, PWA 1480, 001 DIRECTION'.....MATERIAL DESCRIPTION
C 1.54    1.57    1    8...............YIELD & ULTIMATE STRENGTHES, NDIV, NPTS
C 8     -1.0    1......................# PTS IN DIV, STRAIN RATIO, REGION
C 0.89       6800......................S(1) N(1)        RAW
C 0.89      15000......................S(2) N(2)   STRAIN-LIFE
C 0.67      27000......................S(3) N(3)       (S/N)
C 0.67      43200......................S(4) N(4)        DATA
C 0.56     139300......................S(5) N(5)       POINTS
C 0.56     545200......................S(6) N(6)      FOR THE
C 0.56     147000......................S(7) N(7)      SPECIFIC
C 0.39    4344800......................S(8) N(8)      MATERIAL
C 0.00.................................NO VALUE OF So SUPPLIED (%)
C 1  0.................................NUMBER OF REGIONS:W/DATA W/O DATA
C 1.0E+36..............................LIFE BOUNDARIES:  REGION 1
C 0.00.................................CONSTRAINT ON COEFF. OF VARIATION
C 0    0.00     0.00...................0 PTS IN RANGE, LOWER BOUND, UPPER BOUND
C 0.0    0.0    0.0....................NORMAL DIST. PRIORS: DELTA, Mo, SIGMA2
C
C*************************************************************************
C                             LIST OF VARIABLES
```

```
C**********************************************************************
C
C ALLM()      2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C BIGK()      1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C BIGK1       EQUAL TO BIGK(1) — DUMMY PARAMETER FOR CALLS TO SUBROUTINE
C               EXPCTD
C BLDLIF      REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND LCF
C               LIFE CALCULATION
C BLFPER()    1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C BLFPOS()    1-D ARRAY CONTAINING POSITION IN LIFE() OF EMPIRICAL BLIVES
C BZERO       WEIBULL SHAPE PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C DUM         DUMMY VARIABLE
C EBEND       SELECTED VALUE FOR BENDING STRAIN (%)
C EBENDA      EBEND LOWER BOUND
C EBENDB      EBEND UPPER BOUND
C EMNOM       NOMINAL MECHANICAL STRAIN (%)
C EPSL        LOGNORMAL WORST OF NSYM RANDOM VARIATE
C EPSW        WEIBULL WORST OF NSYM RANDOM VARIATE
C ETHNOM()    1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY
C FAA, FAB, FAC, FAD, FAE, FAF
C             COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
C FACTR       SCALE FACTOR EQUAL TO PHI * KRATIO * Z
C FAERRM      STARTUP THERMAL STRAIN RESPONSE SURFACE MEAN
C FAERRS      STARTUP THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
C FD1A, FD1B, FD1C, FD1D, FD1E, FD1F
C             COEFFICIENTS FOR FD1, ONE OF THE DECELERATION FUNCTIONS
C FD2A, FD2B
C             COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS
C FD3A, FD3B
C             COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS
C FDERRM      DECEL THERMAL STRAIN RESPONSE SURFACE MEAN
C FDERRS      DECEL THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
C FIFTY       EQUAL TO .5 — USED TO ACCESS 50% POINT IN LIFEL() AND LIFEW()
C FTU         MATERIAL ULTIMATE STRENGTH (%)
C FTY         MATERIAL YIELD STRENGTH (%)
C HGAS        SELECTED HOT GAS FILM COEFFICIENT, Hgas
C HGASA       HGAS LOWER BOUND
C HGASB       HGAS UPPER BOUND
C HGASR       SELECTED RHO FOR HGAS
C HGASR1      HGAS - RHO LOWER BOUND
C HGASR2      HGAS - RHO UPPER BOUND
C HGAST       SELECTED THETA FOR HGAS
C HGAST1      HGAS - THETA LOWER BOUND
C HGAST2      HGAS - THETA UPPER BOUND
C I           CONTROLS INNER DO LOOP
C IOUT        CONTROLS DUMP TO FILE IOUTPR
C J           CONTROLS DO LOOP FOR EACH BLIFE
C K           CONTROLS OUTER DO LOOP
C KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C L           CONTROLS DO LOOP FOR EACH REGION
C LAMA        SELECTED COEFFICIENT OF THERMAL ACCURACY FACTOR, LAMbda Alpha
C LAMAA       LAMA LOWER BOUND
C LAMAB       LAMA UPPER BOUND
C LAMDA       SELECTED DAMAGE ACCUMULATION MODEL ACCURACY FACTOR, LAMbda
C               Damage Accumulation
C LAMDAA      LAMDA LOWER BOUND
C LAMDAB      LAMDA UPPER BOUND
C LAMG        SELECTED UNCERTAINTY IN Tgas
C LAMGA       LAMG LOWER BOUND
C LAMGB       LAMG UPPER BOUND
C LAMGR       SELECTED RHO FOR LAMG
C LAMGR1      LAMG - RHO LOWER BOUND
C LAMGR2      LAMG - RHO UPPER BOUND
C LAMGT       SELECTED THETA FOR LAMG
C LAMGT1      LAMG - THETA LOWER BOUND
C LAMGT2      LAMG - THETA UPPER BOUND
C LAMP        SELECTED DEVIATION IN BLADE PULL DUE TO BLADE MASS, LAMbda Pull
C LAMPA       LAMP LOWER BOUND
C LAMPB       LAMP UPPER BOUND
C LAMTM       SELECTED TMF MODEL ACCURACY FACTOR, LAMbda TMf
C LAMTMA      LAMTM LOWER BOUND
C LAMTMB      LAMTM UPPER BOUND
C LIFEL()     1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
C               USING THE LOGNORMAL DISTRIBUTION — SORTED VALUES OF THE
```

```
C                 LEFT-HAND TAIL
C LIFEW()       1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
C                 USING THE WEIBULL DISTRIBUTION - SORTED VALUES OF THE
C                 LEFT-HAND TAIL
C LIFL          MISSIONS TO FAILURE BASED ON EPSL
C LIFW          MISSIONS TO FAILURE BASED ON EPSW
C LNA()         1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
C LNZ           NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
C LPHIM()       1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
C MANAL         SELECTED MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
C MANALA        MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR LOWER BOUND
C MANALB        MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR UPPER BOUND
C MAXBLF        MAXIMUM NUMBER OF BLIVES TO BE PROVIDED
C MAXDAT        MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED
C MAXLIF        MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA,
C                 ALPHA CALCULATION
C MAXM          MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY
C MAXMM         MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION
C MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C MCOUNT        NUMBER OF M's TO BE USED TO CALCULATE MEDIAN S/N CURVE
C MEDKB()       1-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION
C                 (BOOTSTRAP OPTION)
C MEDM()        1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C MEDMB()       1-D ARRAY CONTAINING THE MEAN M VALUES FOR EACH REGION
C                 (BOOTSTRAP OPTION)
C MID           POINTER TO THE MEDIAN M VALUES - EQUAL TO HALF OF MCOUNT
C MM()          1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C MODER1        MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
C MODER2        MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE
C MPROC         Materials PROcess variation - CONTROLS MATERIALS PROCESS
C                 VARIATION - 0 - NO VARIATION;  1 - VARIATION
C MU()          1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION MEAN FOR EACH REGION
C NBLIFE        NUMBER OF BLIVES TO BE PROVIDED
C NBND()        1-D ARRAY CONTAINING UPPER BOUNDS FOR THE NUMREG LIFE REGIONS
C                 OF INTEREST FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
C NEWLIF        LIFE VALUE RETURNED FROM CALL TO BLDLIF
C NF()          2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL S/N DATA
C                 SET BROKEN INTO LIFE REGIONS
C NHYPER        SIZE OF OUTER LOOP
C NLIFE         SIZE OF INNER LOOP
C NLIFET        TOTAL NUMBER OF LIVES CALCULATED BY PFM
C NMED          CONTROLS MEDIAN CALCULATION FOR THE NORMAL DISTRIBUTION CASE -
C                 0 - NO MEDIAN CALCULATION;  1 - MEDIAN CALCULATION DESIRED
C NOMSPD        NOMINAL ROTOR SPEED, RPM
C NPTS()        1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION FOR
C                 THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
C NSYM          SYMMETRY NUMBER
C NTIME         NUMBER OF POINTS IN STRAIN-TIME HISTORY
C NUMREG        NUMBER OF REGIONS OF INTEREST
C PERIOD        LENGTH OF TIME IN MISSIONS OF TIME HISTORY
C PHI           WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C PSIG          EQUAL TO SQRT(PVAR) - MATERIALS PROCESS STANDARD DEVIATION
C PVAR          MATERIALS PROCESS VARIATION
C RAND          RANDOM NUMBER SEED
C RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR
C                 EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L)
C                 IS THE UPPER BOUND
C RESID()       1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH
C                 POINT IN THE SPECIFIC MATERIAL S/N DATA SET
C RPM()         1-D ARRAY CONTAINING ROTOR SPEED HISTORY (rpm)
C SBND()        1-D ARRAY CONTAINING THE STRAIN VALUES (%, R = -1.0)
C                 CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C                 REGION CONTAINED IN NBND()
C SIG()         1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C SLOPE         SELECTED DECELERATION SLOPE, m (deg R / sec)
C SLOPEA        m LOWER BOUND
C SLOPEB        m UPPER BOUND
C SLOPR         SELECTED RHO FOR m
C SLOPR1        m - RHO LOWER BOUND
C SLOPR2        m - RHO UPPER BOUND
C SLOPT         SELECTED THETA FOR m
C SLOPT1        m - THETA LOWER BOUND
C SLOPT2        m - THETA UPPER BOUND
```

```
C SPEED       SELECTED STEADY STATE ROTOR SPEED, RPM
C SPEEDM      MEAN OF ROTOR SPEED (MU, NORMAL DISTRIBUTION)
C SPEEDS      STANDARD DEVIATION OF ROTOR SPEED (SIGMA, NORMAL DISTRIBUTION)
C STR()       2-D ARRAY CONTAINING STRAIN POINTS (STRAIN RATIO = -1.0) FOR
C               THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS
C SZERO       STRAIN TENSILE TEST POINT, So
C TANAL       SELECTED THERMAL STRAIN ANALYSIS ACCURACY FACTOR
C TANALA      THERMAL STRAIN ANALYSIS ACCURACY FACTOR LOWER BOUND
C TANALB      THERMAL STRAIN ANALYSIS ACCURACY FACTOR UPPER BOUND
C TGAS        SELECTED GAS TEMPERATURE Tgas
C TGASA       GAS TEMPERATURE LOWER BOUND
C TGASB       GAS TEMPERATURE UPPER BOUND
C TGASR       SELECTED RHO FOR GAS TEMPERATURE
C TGASR1      GAS TEMPERATURE - RHO LOWER BOUND
C TGASR2      GAS TEMPERATURE - RHO UPPER BOUND
C TGAST       SELECTED THETA FOR GAS TEMPERATURE
C TGAST1      GAS TEMPERATURE - THETA LOWER BOUND
C TGAST2      GAS TEMPERATURE - THETA UPPER BOUND
C TRBIGK()    1-D ARRAY CONTAINING VALUES OF BIGK() CORRECTED FOR SZERO,
C               PHI, KRATIO, AND Z
C TRSBND()    1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR
C               EACH REGION CALCULATED FOR EACH TRIAL
C TRUNC       VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)
C TSTART      STARTING DECELERATION TEMPERATURE (deg R)
C TSTMU       MEAN OF TSTART
C TSTSIG      STANDARD DEVIATION OF TSTART
C TSUBI       THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS
C VARY        CONTROLS TYPE OF CURVE VARIATION DESIRED — 0 - NO VARIATION;
C               1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;  3 -
C               TRUNCATED NORMAL VARIATION
C WEXP        WALKER EXPONENT
C Z           LOGNORMAL(0,PVAR) GENERATED RANDOM VARIATE
C ZROREG      ZeRO REGion — VALUES CHOSEN TO FACILITATE REGION DO LOOP
C               BEGINNING VALUE — 0 ZERO REGION EXISTS, 1 - NO ZERO REGION


C************************************************************************


C   FUNCTION BLDLIF PERFORMS THE DRIVER TRANSFORMATION AND CALLS RAINF3
C   TO CALCULATE THE FATIGUE LIFE
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  7JAN92      COMMENTS:  17APR92
C      VERSION:  BLDLCF 3.4 (MATCHR V8.5, RAINF3 V1.1)
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.

      FUNCTION BLDLIF (TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF,
     &                 MODER1, RPM, TSUBI, SPEED, SLOPE, TSTART, FD1A,
     &                 FD1B, FD1C, FD1D, FD1E, FD1F, MODER2, FD2A,
     &                 FD2B, FD3A, FD3B, ETHNOM, MANAL, LAMP, NOMSPD,
     &                 EMNOM, TANAL, LAMA, LAMG, EBEND, NTIME, TRUNC,
     &                 PERIOD, WEXP, MM, LNA, LPHIM, KRATIO, LNZ,
     &                 TRSBND, ZROREG, NUMREG, SZERO)

C   SUBPROGRAMS:  RAINF3
C        INPUTS:  TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF, MODER1, RPM,
C                 TSUBI, SPEED, SLOPE, TSTART, FD1A, FD1B, FD1C, FD1D,
C                 FD1E, FD1F, MODER2, FD2A, FD2B, FD3A, FD3B, ETHNOM, MANAL,
C                 LAMP, NOMSPD, EMNOM, TANAL, LAMA, LAMG, EBEND, NTIME,
C                 TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO, LNZ, TRSBND,
C                 ZROREG, NUMREG, SZERO
C       OUTPUTS:  BLDLIF

C     IMPLICIT NONE

      INTEGER  MAXM, MAXREG

      PARAMETER (MAXM = 50, MAXREG = 3)
```

```
      COMMON    IOUT

      INTEGER   I, IOUT, NTIME, NUMREG, TSUBI, ZROREG

      REAL      BLDLIF, EBEND, EM(MAXM), EMNOM, ETH(MAXM), ETHNOM(MAXM),
     &          ETOT(MAXM), FA, FAA, FAB, FAC, FAD, FAE, FAF, FD1,
     &          FD1A, FD1B, FD1C, FD1D, FD1E, FD1F, FD2, FD2A, FD2B,
     &          FD3, FD3A, FD3B, HGAS, KRATIO, LAMA, LAMG, LAMP,
     &          LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MANAL,
     &          MM(0:MAXREG), MODER1, MODER2, NOMSPD, PERIOD, RAINF3,
     &          RPM(MAXM), SLOPE, SPEED, SZERO, TANAL, TGAS,
     &          TRSBND(0:MAXREG), TRUNC, TSTART, WEXP

C                             LIST OF VARIABLES
C
C EBEND        SELECTED VALUE FOR BENDING STRAIN (%)
C EM()         1-D ARRAY CONTAINING THE SIMULATED MECHANICAL STRAIN-TIME
C                HISTORY (%)
C EMNOM        NOMINAL MECHANICAL STRAIN (%)
C ETH()        1-D ARRAY CONTAINING THE SIMULATED THERMAL STRAIN-TIME HISTORY
C ETHNOM()     1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY
C ETOT()       1-D ARRAY CONTAINING THE TOTAL STRAIN-TIME HISTOY
C FA           VALUE OF ACCELERATION FUNCTION FOR THERMAL STRAIN - SECOND
C                ORDER POLYNOMIAL AS A FUNCTION OF TGAS AND HGAS
C FAA, FAB, FAC, FAD, FAE, FAF
C              COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
C FD1          VALUE OF DECELERATION FUNCTION FOR THERMAL STRAIN - SECOND
C                ORDER POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE
C FD1A, FD1B, FD1C, FD1D, FD1E, FD1F
C              COEFFICIENTS FOR FD1, ONE OF THE DECELERATION FUNCTIONS
C FD2          VALUE OF DECELERATION FUNCTION FOR TIME - SECOND ORDER
C                POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE
C FD2A, FD2B
C              COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS
C FD3          VALUE OF DECELERATION FUNCTION FOR ROTOR SPEED - FIRST
C                ORDER POLYNOMIAL (LINEAR) FUNCTION OF TIME
C FD3A, FD3B
C              COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS
C HGAS         SELECTED HOT GAS FILM COEFFICIENT, Hgas
C I            CONTROLS DO LOOP FOR EACH POINT IN TIME HISTORY
C IOUT         CONTROLS DUMP TO FILE IOUTPR
C KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C LAMA         SELECTED VALUE FOR COEFFICIENT OF THERMAL EXPANSION ACCURACY
C                FACTOR, LAMbda Alpha
C LAMG         THE UNCERTAINTY IN Tgas
C LAMP         SELECTED VALUE FOR DEVIATION IN BLADE PULL DUE TO BLADE MASS,
C                LAMbda Pull
C LNA()        1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
C LNZ          NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
C LPHIM()      1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
C MANAL        SELECTED VALUE FOR MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
C MAXM         MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY
C MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C MM()         1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C MODER1       MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
C MODER2       MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE
C NOMSPD       NOMINAL ROTOR SPEED, RPM
C NTIME        NUMBER OF POINTS IN STRAIN-TIME HISTORY
C NUMREG       NUMBER OF REGIONS OF INTEREST
C PERIOD       LENGTH OF TIME IN MISSIONS OF TIME HISTORY
C RAINF3       REAL FUNCTION PERFORMING RAINFlow COUNTING, DAMAGE ACCUMU-
C                LATION AND FATIGUE LIFE PREDICTION (USING THE MATERIALS
C                CHARACTERIZATION MODEL)
C RPM()        1-D ARRAY CONTAINING ROTOR SPEED HISTORY
C SLOPE        SELECTED VALUE FOR DECELERATION SLOPE, deg R / sec
C SPEED        SELECTED VALUE FOR STEADY STATE ROTOR SPEED, rpm
C SZERO        STRAIN TENSILE TEST POINT, So
C TANAL        SELECTED VALUE FOR THERMAL STRAIN ANALYSIS ACCURACY FACTOR
C TGAS         SELECTED VALUE FOR HOT GAS TEMPERATURE Tgas (deg R)
C TRSBND()     1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR
C                EACH REGION CALCULATED FOR EACH TRIAL
C TRUNC        VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)
C TSTART       STARTING DECELERATION TEMPERATURE (deg R)
C TSUBI        THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS
```

```
C WEXP        WALKER EXPONENT
C ZROREG      ZeRO REGion — VALUES CHOSEN TO FACILITATE REGION DO LOOP
C               BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO ZERO
C               REGION


C ** CALCULATE STRAIN HISTORY

      FA = FAA + FAB * TGAS + FAC * HGAS + FAD * TGAS ** 2
     &     + FAE * HGAS ** 2 + FAF * TGAS * HGAS + MODER1
      ETHNOM(1) = FA

      RPM(TSUBI) = SPEED

      FD1 = FD1A + FD1B * TSTART + FD1C * SLOPE + FD1D * TSTART ** 2
     &      + FD1E * SLOPE ** 2 + FD1F * TSTART * SLOPE + MODER2
      FD2 = FD2A + (TSTART - FD2B) / SLOPE
      FD3 = FD3A + FD3B * FD2
      RPM(NTIME) = FD3
      ETHNOM(NTIME) = FD1

      DO 100 I = 1, NTIME
         EM(I) = MANAL * LAMP * (RPM(I) / NOMSPD) ** 2 * EMNOM
         ETH(I) = TANAL * LAMA * ETHNOM(I)
         IF ((I .GT. 1) .AND. (I .LT. TSUBI))
     &      ETH(I) = LAMG * ETH(I)
         ETOT(I) = EBEND + EM(I) + ETH(I)
  100 CONTINUE

      IF (IOUT .EQ. 15) THEN
         WRITE(8,*) 'FA = ', FA, ' ETHNOM1 = ', ETHNOM(1)
         WRITE(8,*) 'RPMI = ', RPM(TSUBI), ' LAMG = ', LAMG
         WRITE(8,*) 'FD1 = ', FD1, ' FD2 = ', FD2
         WRITE(8,*) 'FD3 = ', FD3, ' RPM = ', RPM(NTIME)
         WRITE(8,*) '   ETHNOM = ', ETHNOM(NTIME)
         DO 125 I = 1, NTIME
            WRITE(8,*) 'I = ', I, ' EM = ', EM(I)
            WRITE(8,*) 'ETH = ', ETH(I), ' ETOT = ', ETOT(I)
  125    CONTINUE
      ENDIF

C ** CALL RAINF3 TO CALCULATE DAMAGE AND RESULTING FATIGUE LIFE

      BLDLIF = RAINF3 (ETOT, NTIME, TRUNC, PERIOD, WEXP, MM, LNA,
     &                 LPHIM, KRATIO, LNZ, TRSBND, ZROREG, NUMREG,
     &                 SZERO)


      RETURN
      END


C***********************************************************************


C  SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20JUL90
C     VERSION:  2.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

C  INPUTS:   NEWLIF, LIFE, NLIFET
C  OUTPUTS:  LIFE

C     IMPLICIT NONE

      INTEGER  MAXLIF
```

```
      PARAMETER (MAXLIF = 10000)

      COMMON   IOUT

      INTEGER  I, IOUT, NLIFET, NUM, PLACE

      REAL     LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

C                   LIST OF VARIABLES
C
C  I         CONTROLS DO LOOP FOR INSERTION
C  IOUT      OUTPUT DUMP CONTROLLER
C  LIFE()    1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
C              PFM TO BE SORTED
C  MAXLIF    MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
C              CALCULATION
C  NEWLIF    LIFE VALUE TO BE INSERTED INTO LIFE()
C  NLIFET    TOTAL NUMBER OF LIVES CALCULATED BY PFM
C  NUM       NUMBER OF LIFE VALUES IN LIFE()
C  PLACE     POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
C  TEMP()    1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON
C              INSERTION OF NEWLIF


      NUM = NLIFET / 2

C     FIND POSITION IN LIFE() FOR NEWLIF

      IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

      DO 100 I = 1, NUM
         IF (NEWLIF .LT. LIFE(I)) THEN
            PLACE = I
            GOTO 110
         ENDIF
  100 CONTINUE
  110 CONTINUE

C     STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()

      DO 200 I = (PLACE + 1), NUM
         TEMP(I) = LIFE(I-1)
  200 CONTINUE

C     INSERT NEWLIF

      LIFE(PLACE) = NEWLIF

C     SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

      DO 300 I = (PLACE + 1), NUM
         LIFE(I) = TEMP(I)
  300 CONTINUE

C     IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN

  400 CONTINUE


      RETURN
      END

C***********************************************************************
C   SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C   INDEPENDENT RANDOM VARIATES
C   PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C         DATE:  9MAR87
C   SUBPROGRAM:  RANDOM
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
```

```
C  is acknowledged.
C***********************************************************************

      SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, RHO1, RHO2, THE1, THE2, X, Y

      INTEGER IOUT

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      X = FRAC * (RHO2 - RHO1) + RHO1

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      Y = FRAC * (THE2 - THE1) + THE1

      IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ' RHO2 =', RHO2,
     &         ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y

      RETURN
      END


C***********************************************************************
C  THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  9MAR87
C  SUBPROGRAM:  GAM
C
C  The random variates are generated using the method described in:
C  Johnson, N. L., and Kotz, S., Distribution in Statistics: Continuous
C    Univariate Distributions - 1, Houghton Mifflin Company, 1970,
C    pp. 181-182.
C***********************************************************************

      SUBROUTINE BETAGN (RAND, RHO, THETA, A, B, X)

      COMMON IOUT

      DOUBLE PRECISION RAND

      REAL    A, B, GAM, RHO, THETA, W, X, Y1, Y2

      INTEGER IOUT

      IF (IOUT .EQ. 15) WRITE(8,*) 'RAND =', RAND, ' RHO =', RHO,
     &          ' THETA =', THETA, ' A =', A, ' B =', B, ' X =', X
      Y1 = GAM((RHO * THETA + 1.), RAND)
      Y2 = GAM(((1. - RHO) * THETA + 1.), RAND)
      W = Y1 / (Y1 + Y2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'Y1 =', Y1, ' Y2 =', Y2, ' W =', W

C  TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

      X = W * (B - A) + A
      IF (IOUT .EQ. 15) WRITE(8,*) 'W =', W, ' X =', X

      RETURN
      END
C***********************************************************************

C  The random variates are generated using an "Acceptance/Rejection Method"
C  Fishman, George S., "Sampling From the Gamma Distribution on a
C    Computer," Communications of the ACM, Volume 19, Number 7, July 1976,
C    pp. 407-409.
```

```
      REAL FUNCTION GAM (ALPHA, RAND)

C     SUBPROGRAM:  RANDOM

      COMMON  IOUT

      INTEGER IOUT

      REAL    A, ALPHA, ARG, U1, U2, V1, V2

      DOUBLE PRECISION RAND

      A = ALPHA - 1.
C     IF (IOUT .EQ. 15) WRITE(8,*) 'A =', A, ' ALPHA =', ALPHA

   10 CALL RANDOM (U1, RAND)
      CALL RANDOM (U2, RAND)
      V1 = - ALOG(U1)
      V2 = - ALOG(U2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =',
C    &                  V1, ' V2 =', V2
      ARG = A * (V1 - ALOG(V1) - 1.)
      IF (V2 .LT. ARG) GOTO 10

      GAM = ALPHA * V1
C     IF (IOUT .EQ. 15) WRITE(8,*) 'GAMMA =', GAM

      RETURN
      END


C***********************************************************************


C  SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
C  AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C  FOR THE STRESS FORMULATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  30NOV90       FORMAT/COMMENTS:  15JAN92
C     VERSION:  MATCHR VB1.2, VB1.3
C               MATGRM VB1, VB1.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.


      SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG,
     &                   NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC,
     &                   KRATIO, PVAR, MEDMB, MEDKB, RESID)

C        INPUTS:  READS DATA FROM SPECFD AND RELATD; VARY, MPROC
C       OUTPUTS:  RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG, NUMREG,
C                 NBND, STR, FTUZ, FTYZ, KRATIO, PVAR, MEDMB, MEDKB,
C                 RESID
C   SUBPROGRAMS:  INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDREG, CONCAV,
C                 MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR, EXPB
C         FILES:  5:RELATD-OLD; 6:RELATO-NEW

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
     &        NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
     &        NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG

      REAL    BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
     &        FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
```

```
     &           KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &           LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
     &           MCHAT(2, MAXREG), MEDKB(0:MAXREG), MEDM(MAXREG),
     &           MEDMB(0:MAXREG),  MO(MAXREG), MU(MAXREG),
     &           MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &           PVAR, RANGEM(2, MAXREG), RATSTR(MAXDAT, 0:MAXSET),
     &           RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
     &           RESID(MAXDAT), SIG(MAXREG), SIGMA2(MAXREG),
     &           STR(MAXDAT, MAXREG), SUHAT2(MAXREG), SWHAT2(MAXREG),
     &           SX2(MAXREG), SXY(MAXREG), SY2(MAXREG), SZERO


C                      LIST OF VARIABLES
C
C  BIGKHT      EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
C  BZERO       VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE S/N
C                DATA SET
C  CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L), THE SLOPE OF THE
C                REGRESSION, FOR EACH REGION
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
C                AND SIG() CALCULATION
C  FTUZ        ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  IOUT        OUTPUT DUMP CONTROLLER
C  IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                FOR EACH REGION
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LAMN        LAMBDA-N — RATIO OF Var(Ln N given S) / (m**2 C**2),
C                CONSTANT OVER REGIONS AND COMPONENTS
C  LNNF()      3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()     3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                — MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                BOUND
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
C                MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MC() FOR EACH REGION
C  MEDKB()     1-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION
C                (BOOTSTRAP OPTION)
C  MEDM()      1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C  MEDMB()     1-D ARRAY CONTAINING THE MEAN M VALUES FOR EACH REGION
C                (BOOTSTRAP OPTION)
C  MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C  MPROC       Materials PROcess variation —CONTROLS MATERIALS PROCESS
C                VARIATION — 0 - NO VARIATION;  1 - VARIATION
C  MU()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                SET IN EACH REGION
C  NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                ALL DATA SETS IN A REGION (Number of Points Per Region)
```

```
C  NPTS()       1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  PVAR         MATERIALS PROCESS VARIATION
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  RATSTR()     2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
C                 STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C  RAWNF()      2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                 DATA SETS
C  RAWSTR()     2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
C                 DATA (%) FOR ALL S/N DATA SETS
C  REFNP()      1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C  RESID()      1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH
C                 POINT IN THE SPECIFIC MATERIAL S/N DATA SET
C  SIG()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C  SIGMA2()     1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 VARIANCE FOR EACH REGION
C  STR()        2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SUHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SX2()        1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                 (X = Ln S)
C  SXY()        1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR EACH
C                 REGION (X = Ln S, Y = Ln N)
C  SY2()        1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                 (Y = Ln N)
C  SZERO        STRESS TENSILE TEST POINT, So
C  VARY         CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO
C                 VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                 VARIATION;  3 - TRUNCATED NORMAL VARIATION;  4 - BOOTSTRAP
C  ZROREG       ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                 REGION

      OPEN(5, FILE = 'RELATD', STATUS = 'OLD')
      OPEN(6, FILE = 'RELATO', STATUS = 'NEW')

C  RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C  RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET
C  INFORMATION

C  PERFORM CALCULATIONS COMMON TO UNIFORM, NORMAL, AND BOOTSTRAP
C  TYPE OF VARIATION

C  INITIALIZE PRIMARY ARRAYS

      CALL INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
     &           NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

C  READ, CONVERT, ECHO INFORMATION

      CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR,
     &          LNNF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, NNODAT,
     &          NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO,
     &          SIGMA2, KRATIO, LAMN)

C  CALCULATE RESIDUAL VARIANCES

      CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD,
     &             SWHAT2, SUHAT2, NPPR, MEDMB, MEDKB, RESID)

C  CALCULATE M CONTRAINT BASED ON Co

      CALL FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
```

```
      IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1) .OR. (VARY .EQ. 2)) THEN
C  CALCULATIONS FOR ALL TYPES OF VARIATION SAVE NORMAL
C  CALCULATE BOUNDS FOR CONFIDENCE INTERVALS

         CALL INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                JZERO, MCHAT)

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C  OBTAIN POSTERIOR RANGES ON M

         CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT,
     &                RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

         CALL ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

         CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

         WRITE(7,900)

         DO 25 L = 1, NUMREG
            WRITE(7,905) L, IZERO(1, L), IZERO(2, L),
     &                      JZERO(1, L), JZERO(2, L)
   25    CONTINUE

         WRITE(7,910)

         DO 50 L = 1, NUMREG
            WRITE(7,915) L, MCHAT(2,L), MCHAT(1,L)
   50    CONTINUE

         IF (CZERO .GT. 0.0) THEN
            WRITE(7,960)
            DO 150 L = 1, NUMREG
               IF (MCPNT(L) .EQ. 1) THEN
                  WRITE(7,965) L, MC(1,L)
               ELSEIF (MCPNT(L) .EQ. 2) THEN
                  WRITE(7,970) L, MC(1,L), MC(2,L)
               ENDIF
  150       CONTINUE
         ENDIF

         WRITE(7,920)
         WRITE(7,930)

         DO 100 L = 1, NUMREG
            WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
  100    CONTINUE

         WRITE(7,950)

C  CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO

         CALL MEDIAN (NUMREG, RANGEM, MEDM)

C  CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS

         CALL EXPCTD (1, MEDM, REFNP, STR, NF, SZERO, NUMREG, ZROREG,
     &                NBND, BIGKHT, BZERO)

C  CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED
```

```
         IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
            DO 200 L = 1, NUMREG
               RANGEM(1,L) = MEDM(L)
               RANGEM(2,L) = MEDM(L)
  200       CONTINUE
         ENDIF

      ELSEIF (VARY .EQ. 3) THEN

C  NORMAL VARIATION IS DESIRED

C  CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION

         CALL MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO,
     &               SIGMA2, MCHAT, MU, SIG)

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M

         CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

         CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO,
     &                MPNT, MO, SIGMA2)

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

         CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

         WRITE(7,975)

         DO 350 L = 1, NUMREG
            WRITE(7,980) L, MCHAT(1,L)
  350    CONTINUE

         IF (CZERO .GT. 0.0) THEN
            WRITE(7,960)
            DO 360 L = 1, NUMREG
               IF (MCPNT(L) .EQ. 1) THEN
                  WRITE(7,965) L, MC(1,L)
               ELSEIF (MCPNT(L) .EQ. 2) THEN
                  WRITE(7,970) L, MC(1,L), MC(2,L)
               ENDIF
  360       CONTINUE
         ENDIF

         WRITE(7,920)
         WRITE(7,930)

         DO 370 L = 1, NUMREG
            WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
  370    CONTINUE

         WRITE(7,950)

         WRITE(7,985)
         DO 380 L = 1, NUMREG
            WRITE(7,990) L, MU(L), SIG(L)
  380 CONTINUE

      ELSE

C        BOOTSTRAPPING IS REQUIRED

         WRITE(7,900)
```

```
C         FIRST CALCULATE OTHER REGION PARAMETERS BASED ON THE EXPECTED
C         M AND K VALUES

          CALL EXPB (MEDMB, MEDKB, SZERO, NUMREG, ZROREG, NBND)
      ENDIF

C  PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS

      IF (MPROC .EQ. 1) THEN
         WRITE(7,995) PVAR
      ENDIF


C  FORMAT STATEMENTS

  900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &         'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &         'NASA Contract NAS7-918 is acknowledged.',////,
     &        2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
     &        ///,2X,'95% CONFIDENCE INTERVALS ON C AND m ',
     &         'FOR EACH REGION',/)

  905 FORMAT(7X,'REGION:  ',I1,7X,'Io = (',F12.9,',',F12.9,')',
     &         /,24X,'Jo = (',F12.9,',',F12.9,')')

  910 FORMAT(//,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
     &         //,7X,'REGION',8X,'E(C)',12X,'E(m)',/)

  915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)

  920 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGES ON m FOR EACH '
     &         'REGION')

  930 FORMAT(//,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

  940 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

  950 FORMAT(///)

  960 FORMAT(//,2X,'RANGE ON m FOR EACH REGION IMPLIED BY C '
     &         'CONSTRAINT',
     &         //,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

  965 FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')

  970 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

  975 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &         'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &         'NASA Contract NAS7-918 is acknowledged.',////,
     &        2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
     &        ///,2X,'ESTIMATES OF m FOR EACH REGION',
     &        //,7X,'REGION',12X,'E(m)',/)

  980 FORMAT(9X,I1,11X,F10.6)

  985 FORMAT(2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS',
     &         //,2X,'REGION',5X,'MEAN',8X,'STD DEV',/)

  990 FORMAT(5X,I1,5X,F7.4,5X,E11.5)

  995 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ',
     &        'MEDIAN S/N CURVE',/,2X,'WARRANTED BY THE AVAILABLE ',
     &        'INFORMATION',//,7X,E11.5)

      RETURN
      END


C***********************************************************************
```

```
C  SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C  ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  5OCT87
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRMNAT

      WRITE(8,*) 'PROGRAM EXECUTION TERMINATED'
      STOP
      END


C************************************************************************


C  SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C  USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR,
     &                 REFNP, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

C  INPUTS:   —
C  OUTPUTS:  NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
C            NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
     &        NPTS(0:MAXSET), REFNP(MAXREG)

      REAL    DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
     &        RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG)

C                    LIST OF VARIABLES
C
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                SIG() CALCULATION
C  I           CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT        OUTPUT DUMP CONTROLLER
C  J           CONTROLS DO LOOP FOR EACH DATA SET
C  K           CONTROLS DO LOOP FOR EACH POINT IN A REGION
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LNNF()      3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()     3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
```

```
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                IN EACH REGION
C  NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  RATSTR()    2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
C                STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C  RAWNF()     2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                DATA SETS
C  RAWSTR()    2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OF TOTAL STRAIN
C                DATA (%) FOR ALL S/N DATA SETS
C  REFNP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C  SIGMA2()    1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                VARIANCE FOR EACH REGION
C  STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

      DO 100 J = 0, MAXSET
         NPTS(J) = 0.0
  100 CONTINUE

      DO 200 L = 1, MAXREG
         DO 250 J = 0, MAXSET
            NP(J, L) = 0.0
  250    CONTINUE
  200 CONTINUE

      DO 300 J = 0, MAXSET
         DO 350 I = 1, MAXDAT
            RAWNF(I,J) = 0.0
            RAWSTR(I,J) = 0.0
            RATSTR(I,J) = 0.0
  350    CONTINUE
  300 CONTINUE

      DO 400 L = 1, MAXREG
         DO 425 K = 1, MAXDAT
            DO 450 J = 0, MAXSET
               LNNF(K,J,L) = 0.0
               LNSTR(K,J,L) = 0.0
  450       CONTINUE
  425    CONTINUE
  400 CONTINUE

      DO 500 L = 1, MAXREG
         DO 550 K = 1, MAXDAT
           NF(K,L) = 0.0
           STR(K,L) = 0.0
  550    CONTINUE
  500 CONTINUE

      DO 600 L = 1, MAXREG
         REFNP(L) = 0
         MPNT(L) = 0
         MZERO(1,L) = 0.0
         MZERO(2,L) = 0.0
         DELTA(L) = 0.0
         MO(L) = 0.0
         SIGMA2(L) = 0.0
  600 CONTINUE

      RETURN
      END


C*************************************************************************


C  SUBROUTINE RCE "READS" THE DATA FROM SPECFD AND RELATD; "CONVERTS"
C  THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO
C  SPECFO AND RELATO.  RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS
C  SPECIFIED BY USER
```

```
C  PROGRAMMER:  L. NEWLIN
C        DATE:  21JUN88     FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP,
     &                LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
     &                NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
     &                FTUZ, FTYZ, DELTA, MO, SIGMA2, KRATIO, LAMN)

C  INPUTS:    VARY, MPROC
C  OUTPUTS:   NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP,
C             STR, NF, SZERO, ZROREG, NUMREG, NNODAT, NSETS, NBND,
C             CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2,
C             KRATIO, LAMN
C  SUBPROGRAMS:  TRMNAT, CONVRT

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON IOUT

      INTEGER COUNT, I, IOUT, J, K, L, M, MPNT(MAXREG), MPROC, NDIV,
     &        NNODAT, NP(0:MAXSET, MAXREG), NPTS(0:MAXSET), NSETS,
     &        NUM, NUMREG, REFNP(MAXREG), REG, VARY, ZROREG

      REAL    CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTYZ,
     &        KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        RATIO, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG), SZERO

      CHARACTER*40  DESCRP(0:MAXSET)


C                         LIST OF VARIABLES
C
C  COUNT      INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO,
C               CONVERSION, AND BREAK UP
C  CZERO      EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C               COEFFICIENT OF VARIATION, Co
C  DELTA()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C               SIG() CALCULATION
C  DESCRP()   1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
C  FTU        ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
C  FTUZ       ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTY        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ       YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  J          CONTROLS DO LOOP FOR EACH DATA SET
C  K          CONTROLS DO LOOP FOR EACH POINT IN A REGION
C  KRATIO     RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LAMN       LAMBDA-N - RATIO OF Var (Ln N given S) / (m**2 C**2),
C               CONSTANT OVER ALL REGIONS AND COMPONENTS
C  LNNF()     3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()    3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  M          CONTROLS DO LOOP FOR EACH DATA DIVISION
C  MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET     MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MO()       1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C               MEAN FOR EACH REGION
C  MPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MZERO() FOR EACH REGION
C  MPROC      Materials PROcess variation - CONTROLS MATERIALS PROCESS
C               VARIATION - 0 - NO VARIATION;  1 - VARIATION
C  MZERO()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C               EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
```

```
C                 IS THE UPPER BOUND
C   NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST
C   NDIV        NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO,
C                 REGION PAIRS DURING INPUT
C   NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                 IN EACH REGION
C   NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C   NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUM         NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   RATIO       STRESS RATIO (R = -1.0 IS DESIRED)
C   RATSTR()    2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
C                 RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C   RAWNF()     2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                 DATA SETS
C   RAWSTR()    2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
C                 DATA (%) FOR ALL S/N DATA SETS
C   REFNP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C   REG         REGION OF INTEREST IN A PARTICULAR DIVISION
C   SIGMA2()    1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 VARIANCE FOR EACH REGION
C   STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SZERO       STRESS TENSILE TEST POINT, So
C   VARY        CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO
C                 VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                 VARIATION;  3 - TRUNCATED NORMAL VARIATION
C   ZROREG      ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                 REGION

C   INITIALIZE COUNT AND NBND()

      COUNT = 0

      DO 10 L = 0, MAXREG
         NBND(L) = 0.0
   10 CONTINUE

C   INPUT DATA ON SPECIFIC MATERIAL FROM SPECFD AND ECHO TO SPECFO

      READ(1,*) DESCRP(0), FTY, FTU, NDIV, NPTS(0)

      IF (NPTS(0) .GT. MAXDAT) THEN
         WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &              'SPECIFIC MATERIAL'
         CALL TRMNAT
      ENDIF

      WRITE(3,900) DESCRP(0), FTY, FTU, NPTS(0)
      IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)

      WRITE(3,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

C     STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ

      FTUZ = FTU
      FTYZ = FTY

C     INPUT STRESS/LIFE INFORMATION - INCLUDING STRESS RATIO AND REGION
C     INFORMATION FROM SPECFD AND ECHO TO SPECFO

      DO 100 M = 1, NDIV

         READ (1,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE FOR RATIO:  ', RATIO
```

```
            CALL TRMNAT
         ENDIF

         IF (REG .GT. MAXREG) THEN
            WRITE(8,*) 'ERROR:  OVER REGION LIMIT IN SPECIFIC DATA SET'
            CALL TRMNAT
         ENDIF

         DO 110 I = (COUNT + 1), (COUNT + NUM)
            READ(1,*) RAWSTR(I,0), RAWNF(I,0)
  110    CONTINUE

C        CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT

         IF (RATIO .EQ. -1.0) THEN

C           STRESS RATIO IS CORRECT

            DO 120 I = (COUNT + 1), (COUNT + NUM)
               RATSTR(I,0) = RAWSTR(I,0)
  120       CONTINUE

         ELSE

C           STRESS RATIO TRANSFORMATION MUST BE DONE

            CALL CONVRT (0, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR,
     &                   RATIO, FTU, FTY)

         ENDIF

C        ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL

         DO 130 I = (COUNT + 1), (COUNT + NUM)

            WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
     &                   RATSTR(I,0), RAWNF(I,0)

            IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
     &                   RATIO, REG, RATSTR(I,0), RAWNF(I,0)

  130    CONTINUE

C        BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C        EXPCTD, AND PAREST

         K = NP(0,REG)

         DO 140 I = (COUNT + 1), (COUNT + NUM)

            K = K + 1
            LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
            LNNF(K,0,REG) = ALOG(RAWNF(I,0))
            STR(K,REG) = RATSTR(I,0)
            NF(K,REG) = RAWNF(I,0)

  140    CONTINUE

         IF (K .GT. MAXDAT) THEN
            WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &                 'SPECIFIC MATERIAL'
            CALL TRMNAT
         ENDIF

         NP(0,REG) = K
         REFNP(REG) = K
         COUNT = COUNT + NUM

  100 CONTINUE

      IF (NPTS(0) .NE. COUNT) THEN
         WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &              'INCORRECTLY SPECIFIED'
         WRITE(8,*) 'IN SPECIFIC DATA SET'
         CALL TRMNAT
```

```
      ENDIF

      READ(1,*) SZERO
      IF (NINT (SZERO) .GT. 0) THEN
         ZROREG = 0
      ELSE
         ZROREG = 1
      ENDIF
      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'SZERO = ', SZERO, ' ZROREG = ', ZROREG

C     INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION

      READ(1,*) NUMREG, NNODAT

      IF ((NUMREG + NNODAT) .GT. MAXREG)  THEN
         WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON NUMBER OF REGIONS'
         CALL TRMNAT
      ENDIF

      DO 150 L = ZROREG, (NUMREG + NNODAT)
         READ(1,*) NBND(L)
  150 CONTINUE

      READ(1,*) CZERO

      DO 160 L = 1, (NUMREG + NNODAT)
         READ(1,*) MPNT(L), MZERO(1,L), MZERO(2,L)
  160 CONTINUE

      WRITE(3,913)
      IF (ZROREG .EQ. 0) WRITE(3,914) SZERO
      IF (IOUT .EQ. 10) THEN
         WRITE(8,913)
         IF (ZROREG .EQ. 0) WRITE(8,914) SZERO
      ENDIF

      WRITE(3,915) NUMREG, NNODAT
      IF (IOUT .EQ. 10) WRITE(8,915) NUMREG, NNODAT

      DO 170 L = ZROREG, (NUMREG + NNODAT)
         WRITE(3,920) NBND(L)
         IF (IOUT .EQ. 10) WRITE(8,920) NBND(L)
  170 CONTINUE

      WRITE(3,925) CZERO
      IF (IOUT .EQ. 10) WRITE(8,925) CZERO

      DO 180 L = 1, (NUMREG + NNODAT)
         WRITE(3,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
         IF (IOUT .EQ. 10)
     &      WRITE(8,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
         IF ((VARY .EQ. 3) .AND. (MPNT(L) .EQ. 0)) THEN
            WRITE(8,*) 'ERROR:  NORMAL VARIATION REQUIRES A PRIOR ',
     &                 'RANGE ON M'
            CALL TRMNAT
         ENDIF
  180 CONTINUE

      IF (VARY .EQ. 3) THEN
C        READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
         WRITE(3,945)
         IF (IOUT .EQ. 10) WRITE(8,945)
         DO 190 L = 1, (NUMREG + NNODAT)
            READ(1,*) DELTA(L), MO(L), SIGMA2(L)
            WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
            IF (IOUT .EQ. 10)
     &         WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
            IF ((DELTA(L) .LT. 0.0) .OR.
     &         ((DELTA(L) .GT. 0.0) .AND. (MO(L) .LE. 0.0))) THEN
               WRITE(8,*) 'ERROR:  BAD VALUE FOR DELTA OR VALUE OF MO ',
     &                    'INCONSISTENT WITH DELTA IN REGION ', L
               CALL TRMNAT
            ENDIF
  190    CONTINUE
```

```
      ENDIF

      IF (MPROC .EQ. 1) THEN
         READ(1,*) KRATIO, LAMN
         WRITE(3,955) KRATIO, LAMN
         IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
      ENDIF

C  BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATD
C  AND THEN ECHO TO RELATO

      READ(5,*) NSETS

      IF (NSETS .GT. MAXSET) THEN
         WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF RELATED DATA SETS'
         CALL TRMNAT
      ENDIF

      WRITE(6,935) NSETS

      DO 200 J = 1, NSETS

         COUNT = 0

         IF (IOUT. EQ. 10) WRITE(8,*)'J =', J, ' NSETS =', NSETS

         READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)

         IF (NPTS(J) .GT. MAXDAT) THEN
            WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF POINTS IN ',
     &                 'SET ', J
            CALL TRMNAT
         ENDIF

         WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
         IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)

         WRITE(6,905)
         IF (IOUT .EQ. 10) WRITE(8,905)

         DO 300 M = 1, NDIV

            READ(5,*) NUM, RATIO, REG

            IF (ABS(RATIO) .GT. 1.0) THEN
               WRITE(8,*) 'ERROR:  INVALID VALUE OF RATIO:  ', RATIO
               CALL TRMNAT
            ENDIF

            IF (REG .GT. MAXREG) THEN
               WRITE(8,*)
     &            'ERROR:  OVER REGION LIMIT IN RELATED MATERIAL ', J
               CALL TRMNAT
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
               WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
            ENDIF

            DO 310 I = (COUNT + 1), (COUNT + NUM)
               READ(5,*) RAWSTR(I,J), RAWNF(I,J)
  310       CONTINUE

C  CHECK IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT

            IF (RATIO .EQ. -1.0) THEN

C              STRESS RATIO IS CORRECT

               DO 320 I = (COUNT + 1), (COUNT + NUM)
                  RATSTR(I,J) = RAWSTR(I,J)
  320          CONTINUE
```

```
            ELSE
C
               STRESS RATIO TRANSFORMATION MUST BE DONE
               CALL CONVRT(J, (COUNT + 1), (COUNT + NUM), RAWSTR,
     &                     RATSTR, RATIO, FTU, FTY)

            ENDIF

C           RECORD BOTH S/N DATA SETS TO RELATO

            DO 330 I = (COUNT + 1), (COUNT + NUM)

               WRITE(6,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG,
     &                      RATSTR(I,J), RAWNF(I,J)

               IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,J), RAWNF(I,J),
     &                      RATIO, REG, RATSTR(I,J), RAWNF(I,J)

  330       CONTINUE

            K = NP(J,REG)

            DO 340 I = (COUNT + 1), (COUNT + NUM)

               K = K + 1
               LNSTR(K,J,REG) = ALOG(RATSTR(I,J))
               LNNF(K,J,REG) = ALOG(RAWNF(I,J))

  340       CONTINUE

            IF (K .GT. MAXDAT) THEN
               WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF POINTS ',
     &                    'IN SET ', J
               CALL TRMNAT
            ENDIF

            NP(J,REG) = K
            COUNT = COUNT + NUM

  300    CONTINUE

         IF (NPTS(J) .NE. COUNT) THEN
            WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &                 'INCORRECTLY SPECIFIED IN SET ', J
            CALL TRMNAT
         ENDIF

  200 CONTINUE


C  FORMAT STATEMENTS USED TO WRITE TO SPECFO AND RELATO

  900 FORMAT(////,13X,'MATERIAL INPUT',///,2X,'DESCRIPTION:',2X,A40,//,
     &        2X,'YIELD STRENGTH',18X,E11.5,//,2X,'ULTIMATE STRENGTH',
     &        15X,E11.5,//,2X,'NUMBER OF POINTS',16X,I2)

  905 FORMAT(//,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
     &        /,5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
     &        'STRESS',7X,'LIFE'/)

  910 FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)

  913 FORMAT(//)

  914 FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN So OF',
     &        5X,E11.5)

  915 FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA  ',
     &        /,2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
     &        /,2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
     &        '(CYCLES): ',/)

  920 FORMAT(10X,E9.3)
```

```
  925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
     &        'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
     &        //,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
     &        //,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',
     &        5X,'UPPER BOUND',/)

  930 FORMAT(6X,I1,11X,I1,12X,F7.4,9X,F7.4)

  935 FORMAT(20X,'NUMBER OF DATA SETS:',2X,I2,//,17X,
     &        'NOTE:  ALL Kt ASSUMED TO BE 1.0',////,23X,
     &        'TRANSFORMED DATA')

  940 FORMAT(///,2X,'DESCRIPTION:',2X,A40,
     &        //,2X,'YIELD STRENGTH',18X,F7.0,
     &        //,2X,'ULTIMATE STRENGTH',15X,F7.0,
     &        //,2X,'NUMBER OF POINTS',16X,I2)

  945 FORMAT(/,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS:',
     &        //,2X,'REGION',5X,'DELTA',8X,'mo',10X,'SIGMA2',/)

  950 FORMAT(5X,I1,5X,F7.2,5X,F7.4,5X,E11.5)

  955 FORMAT(//,2X,'MATERIALS PROCESS VARIATION INFORMATION',
     &        //,2X,'MEDK*/MEDK:',5X,E11.5,/,5X,'LAMBDAN:',5X,E11.5)

      RETURN
      END

C************************************************************************


C   THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C   STRESS RATIO, R, IS NOT -1.0
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                       V8.3, V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)

C   INPUTS:    J, NUM1, NUM2, STR, R, FTU, FTY
C   OUTPUTS:   RSTR

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXSET

      PARAMETER (MAXDAT = 50, MAXSET = 5)

      COMMON IOUT

      INTEGER I, IOUT, J, NUM1, NUM2

      REAL     FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET),
     &         STR(MAXDAT, 0:MAXSET), TEST


C                    LIST OF VARIABLES
C
C   FTU          ULTIMATE STRENGTH OF MATERIAL (PSI)
C   FTY          YIELD STRENGTH OF MATERIAL (PSI)
C   I            CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C   IOUT         OUTPUT DUMP CONTROLLER
C   J            DATA SET OF INTEREST
C   MAXDAT       MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C   MAXSET       MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C   NUM1         FIRST INDEX TO BE TRANSFORMED
C   NUM2         LAST INDEX TO BE TRANSFORMED
C   R            STRESS RATIO (R = -1.0 IS DESIRED)
C   RSTR()       STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
C   STR()        ARRAY CONTAINING STRESS VALUES (PSI) FOR S/N CURVE
```

```
C  TEST        Kt * Smax * (1 - R)/2 , TO BE COMPARED WITH FTY

C  Kt IS ASSUMED TO BE ONE
      DO 100 I = NUM1, NUM2
         TEST = STR(I,J) * (1.0 - R)/2.0
         IF (IOUT.EQ.10) WRITE(8,*) 'I =',I,' J =',J,' TEST =',TEST

         IF (TEST .GE. FTY) THEN
            RSTR(I,J) = TEST
            IF (IOUT.EQ.10) WRITE(8,*)'1:RSTR() =',RSTR(I,J)
         ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN
            RSTR(I,J) = TEST/(1.0 -((FTY - TEST)/FTU))
            IF (IOUT.EQ.10) WRITE(8,*)'2:RSTR() =',RSTR(I,J)
         ELSE
            RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
     &                  /(2.0 * FTU)))
            IF (IOUT.EQ.10) WRITE(8,*)'3:RSTR() =',RSTR(I,J)
         END IF
  100 CONTINUE

      RETURN
      END

C************************************************************************

C  SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C  AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C  X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  15JAN92
C     VERSION:  MATCHR VB1.3   MATGRM VB1.1

      SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY,
     &                   SY2, DD, SWHAT2, SUHAT2, NPPR, MEDMB, MEDKB,
     &                   RESID)

C   INPUTS:  NUMREG, NSETS, NP, LNSTR, LNNF
C  OUTPUTS:  SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR, MEDMB, MEDKB, RESID

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER IOUT, J, K, L, NP(0:MAXSET, MAXREG), NPPR(MAXREG),
     &        NSETS, NUMREG

      REAL    BB(MAXREG), CC(MAXREG), DD(MAXREG),
     &        DIFFX(MAXDAT, 0:MAXSET), DIFFY(MAXDAT, 0:MAXSET),
     &        LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG),
     &        MEANX(0:MAXSET), MEANY(0:MAXSET), MEDKB(0:MAXREG),
     &        MEDMB(0:MAXREG), RESID(MAXDAT), SUHAT2(MAXREG),
```

```
     &         SWHAT2(MAXREG), SX2(MAXREG), SXY(MAXREG), SY2(MAXREG),
     &         WHAT(MAXDAT, 0:MAXSET)

C                 LIST OF VARIABLES
C
C  BB()      1-D ARRAY CONTAINING SXY(L)/SY2(L), THE SLOPE OF THE X ON Y
C              REGRESSION, FOR EACH REGION
C  CC()      1-D ARRAY CONTAINING MEANY-DD(L)*MEANX, THE Y-INTERCEPT OF
C              THE Y ON X REGRESSION, FOR EACH REGION
C  DD()      1-D ARRAY CONTAINING SXY(L)/SX2(L), THE SLOPE OF THE Y ON X
C              REGRESSION, FOR EACH REGION
C  DIFFX()   2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L)
C               AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C  DIFFY()   2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L)
C               AND MEANY(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C  IOUT      OUTPUT DUMP CONTROLLER
C  J         CONTROLS DO LOOP FOR EACH DATA SET
C  K         CONTROLS DO LOOP FOR EACH POINT IN A REGION
C  L         CONTROLS DO LOOP FOR EACH REGION
C  LNNF()    3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()   3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  MAXDAT    MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET    MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MEANX()   1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION
C              L AND DATA SET J (X = Ln S)
C  MEANY()   1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
C              L AND DATA SET J (Y = Ln N)
C  MEDKB()   1-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION
C              (BOOTSTRAP OPTION)
C  MEDMB()   1-D ARRAY CONTAINING THE MEAN M VALUES FOR EACH REGION
C              (BOOTSTRAP OPTION)
C  NP()      2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C              SET IN EACH REGION
C  NPPR()    1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C              ALL DATA SETS IN A REGION (Number of Points Per Region)
C  NSETS     NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  RESID()   1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH
C              POINT IN THE SPECIFIC MATERIAL S/N DATA SET
C  SUHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C              REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C  SWHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C              REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C  SX2()     1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C              (X = Ln S)
C  SXY()     1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
C              EACH REGION (X = Ln S, Y = Ln N)
C  SY2()     1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C              (Y = Ln N)
C  WHAT()    2-D ARRAY CONTAINING THE RESIDUALS OF THE Y ON X REGRESSION
C              (X = Ln S, Y = Ln N)

C  INITIALIZE ARRAYS

      DO 50 L = 1, MAXREG
        SY2(L) = 0.0
        SX2(L) = 0.0
        SXY(L) = 0.0
        SWHAT2(L) = 0.0
        SUHAT2(L) = 0.0
        BB(L) = 0.0
        CC(L) = 0.0
        DD(L) = 0.0
        NPPR(L) = 0
   50 CONTINUE

      DO 55 L = 0, MAXREG
         MEDMB(L) = 0.0
         MEDKB(L) = 0.0
   55 CONTINUE

      DO 60 J = 0, MAXSET
```

```
         DO 70 K = 1, MAXDAT
            DIFFY(K,J) = 0.0
            DIFFX(K,J) = 0.0
            WHAT(K,J) = 0.0
   70    CONTINUE
         MEANY(J) = 0.0
         MEANX(J) = 0.0
   60 CONTINUE

      DO 75 K = 1, MAXDAT
         RESID(K) = 0.0
   75 CONTINUE

C  NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION

      DO 100 L = 1, NUMREG

         DO 200 J = 0, NSETS
C           FIRST CALCULATE SAMPLE X AND Y MEANS
C           FOR DATA SET J IN REGION L
            MEANY(J) = 0.0
            MEANX(J) = 0.0
            IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' J =', J,
     &                       ' NP =', NP(J,L)

            DO 250 K = 1, NP(J,L)
               MEANY(J) = MEANY(J) + LNNF(K,J,L)
               MEANX(J) = MEANX(J) + LNSTR(K,J,L)
               IF (IOUT .EQ. 10) WRITE(8,*) 'LNNF =', LNNF(K,J,L),
     &            ' LNSTR =', LNSTR(K,J,L)
  250       CONTINUE

            MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
            MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
            IF (IOUT .EQ. 10) WRITE(8,*) 'MEANY(J) =', MEANY(J),
     &         ' MEANX(J) =', MEANX(J)

C           NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY,
C           OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
C           DATA SET IN REGION L

            DO 300 K = 1, NP(J,L)
               DIFFY(K,J) = LNNF(K,J,L) - MEANY(J)
               DIFFX(K,J) = LNSTR(K,J,L) - MEANX(J)
               SY2(L) = SY2(L) + DIFFY(K,J) ** 2
               SX2(L) = SX2(L) + DIFFX(K,J) ** 2
               SXY(L) = SXY(L) + DIFFX(K,J) * DIFFY(K,J)
               IF (IOUT .EQ. 10) THEN
                  WRITE(8,*) 'K =', K, ' DIFFY(K,J) =', DIFFY(K,J),
     &                       ' DIFFX(K,J) =', DIFFX(K,J)
                  WRITE(8,*) 'SY2(L) =', SY2(L), ' SX2(L) =', SX2(L),
     &                       ' SXY(L) =', SXY(L)
               ENDIF
  300       CONTINUE

            NPPR(L) = NPPR(L) + NP(J,L) - 1
            IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) =', NPPR(L)
  200    CONTINUE

         IF (SXY(L) .GE. 0.0) THEN
C           LIFE WILL INCREASE WITH INCREASING STRESS - INVALID FOR
C           OUR MODEL
            WRITE(8,*) 'ERROR: SXY >= 0 IN REGION', L
            CALL TRMNAT
         ENDIF

         NPPR(L) = NPPR(L) - 1

         IF (NPPR(L) .LE. 0) THEN
            WRITE(8,*) 'ERROR:  TOO FEW POINTS FOR REGRESSION IN ',
     &                 'REGION ',L
            CALL TRMNAT
         ENDIF

C        CALCULATE THE REGRESSION PARAMETERS
```

```
          SY2(L) = SY2(L) / FLOAT(NPPR(L))
          SX2(L) = SX2(L) / FLOAT(NPPR(L))
          SXY(L) = SXY(L) / FLOAT(NPPR(L))

          DD(L) = SXY(L) / SX2(L)
          BB(L) = SXY(L) / SY2(L)
          CC(L) = MEANY(0) - DD(L) * MEANX(0)
          MEDMB(L) = - DD(L)
          MEDKB(L) = EXP (- CC(L) / DD(L))

          IF (IOUT .EQ. 10) THEN
             WRITE(8,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
     &                  ' SX2(L) =', SX2(L)
             WRITE(8,*) 'SXY(L) =', SXY(L), ' DD(L) =', DD(L),
     &                  ' BB(L) =', BB(L)
             WRITE(8,*) 'CC(L) = ', CC(L), ' MEDMB(L) = ', MEDMB(L),
     &                  ' MEDKB(L) = ', MEDKB(L)
          ENDIF

C         NOW CALCULATE THE RESIDUAL VARIANCES, SWHAT2, SUHAT2, FOR EACH
C         REGION FROM THE Y ON X AND X ON Y REGRESSIONS

          DO 400 J = 0, NSETS
             IF (IOUT .EQ. 10) WRITE(8,*) 'J =', J, ' NP(J,L) =', NP(J,L)

             DO 500 K = 1, NP(J,L)
                WHAT(K,J) = DIFFY(K,J) - DD(L) * DIFFX(K,J)
                SWHAT2(L) = SWHAT2(L) + WHAT(K,J) ** 2
                SUHAT2(L) = SUHAT2(L)
     &                    + (DIFFX(K,J) - BB(L) * DIFFY(K,J)) ** 2
                IF (IOUT .EQ. 10) THEN
                   WRITE(8,*) 'K = ', K, ' WHAT(K,J) = ', WHAT(K,J)
                   WRITE(8,*) 'SWHAT2(L) = ', SWHAT2(L),
     &                        ' SUHAT2(L) = ', SUHAT2(L)
                ENDIF
  500        CONTINUE

  400     CONTINUE

          SWHAT2(L) = SWHAT2(L) / FLOAT(NPPR(L))
          SUHAT2(L) = SUHAT2(L) / FLOAT(NPPR(L))
          IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) =', NPPR(L),
     &       ' SWHAT2(L) =', SWHAT2(L), ' SUHAT2(L) =', SUHAT2(L)

          DO 600 K = 1, NP(0,L)
             RESID(K) = WHAT(K,0) *
     &                  SQRT (FLOAT (NP(0,L)) / FLOAT (NP(0,L)-2))
C             WRITE(4,*) K, RESID(K)
             IF (IOUT .EQ. 10) WRITE(8,*) 'K = ', K, ' RESID = ',
     &          RESID(K), ' WHAT = ', WHAT(K,0)
  600     CONTINUE

  100 CONTINUE

      RETURN
      END


C***********************************************************************


C   SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C   THE Co GIVEN BY THE USER
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  8OCT87      COMMENTS:  13JUL89
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
```

```
C  INPUTS:    NUMREG, CZERO, SX2, SXY, SY2
C  OUTPUTS:   MCPNT, MC

C      IMPLICIT NONE

       INTEGER  MAXREG

       PARAMETER (MAXREG = 3)

       COMMON   IOUT

       INTEGER  IOUT, L, MCPNT(MAXREG), NUMREG

       REAL     ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
      &         SXY(MAXREG), SY2(MAXREG)


C                     LIST OF VARIABLES
C
C  ARG1        INTERMEDIATE CALCULATION VARIABLE
C  ARG2        INTERMEDIATE CALCULATION VARIABLE
C  CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C  CZERO2      EQUAL TO CZERO ** 2
C  IOUT        OUTPUT DUMP CONTROLLER
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
C                CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA - MC(1,L) IS
C                THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
C  MCPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MC() FOR EACH REGION
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  SX2()       1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                (X = Ln S)
C  SXY()       1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR
C                EACH REGION (X = Ln S, Y = Ln N)
C  SY2()       1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                (Y = Ln N)


C      INITIALIZE VARIABLES

       DO 50 L = 1, MAXREG
          MCPNT(L) = 0
          MC(1,L) = 0.0
          MC(2,L) = 0.0
   50  CONTINUE

C      BEGIN CALCULATIONS

       CZERO2 = CZERO ** 2

       IF (IOUT .EQ. 10)
      &   WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2

       DO 100 L = 1, NUMREG

          ARG1 = SX2(L) - CZERO2
          ARG2 = 0.0

          IF (CZERO .EQ. 0.0) THEN

C            THEN NO M CONSTRAINT IS REQUIRED

             MCPNT(L) = 0

          ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN

C            THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M

             MCPNT(L) = 1
             MC(1,L) = - SY2(L) / (2.0 * SXY(L))
```

```
         ELSE
C           THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
C           COMMON CALCULATIONS

            ARG2 = (SXY(L) ** 2 - SY2(L) * ARG1)
            IF (ARG2 .LT. 0.0) THEN
C              ARG2 IS NEGATIVE - IMPLIES M IS COMPLEX
               WRITE(8,*) 'ERROR:  Co TOO LOW'
               CALL TRMNAT
            ELSE
               ARG2 = ARG2 ** 0.5
            ENDIF

            IF (SX2(L) .LT. CZERO2) THEN

C              AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M

               MCPNT(L) = 1
               MC(1,L) = (- SXY(L) - ARG2) / ARG1

            ELSE

C              SX2(L) .GT. CZERO2 - THIS TIME THE M CONSTRAINT IS A RANGE

               MCPNT(L) = 2
               MC(1,L) = (- SXY(L) - ARG2) / ARG1
               MC(2,L) = (- SXY(L) + ARG2) / ARG1

            ENDIF

         ENDIF

  100 CONTINUE

      IF (IOUT .EQ. 10) THEN

         DO 200 L = 1, NUMREG

            WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
            WRITE(8,*) 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)

  200    CONTINUE

      ENDIF

      RETURN
      END

C*************************************************************************


C  SUBROUTINE INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, Io, ON
C  C; AND THE 95% CONFIDENCE INTERVAL, Jo, ON M
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  5OCT87   COMMENTS:  15SEP89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                   JZERO, MCHAT)

C  INPUTS:   NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C  OUTPUTS:  IZERO, JZERO, MCHAT
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE
```

```
      INTEGER CHITAB, MAXREG, TTAB
      PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)
      COMMON  IOUT
      INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG
      REAL    ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
     &        IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
     &        SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,
     &        SX2(MAXREG), T, T025(TTAB)
      DATA (CHI025(I), I = 1, 75) /
     &         0.000982069, 0.506356, 0.215795, 0.484419, 0.831211,
     &         1.237347, 1.68987, 2.17973, 2.70039, 3.24697,
     &         3.81575,  4.40379, 5.00874, 5.62872, 6.26214,
     &         6.90766,  7.56418, 8.23075, 8.90655, 9.59083,
     &         10.28293, 10.9823, 11.6885, 12.4011, 13.1197,
     &         13.8439,  14.5733, 15.3079, 16.0471, 16.7908,
     &         17.53,    18.28,   19.04,   19.80,   20.56,
     &         21.33,    22.10,   22.87,   23.65,   24.4331,
     &         25.21,    25.99,   26.78,   27.57,   28.36,
     &         29.15,    29.95,   30.75,   31.55,   32.3574,
     &         33.16,    33.96,   34.77,   35.58,   36.39,
     &         37.21,    38.02,   38.84,   39.66,   40.4817,
     &         41.30,    42.12,   42.95,   43.77,   44.60,
     &         45.43,    46.26,   47.09,   47.92,   48.7576,
     &         49.59,    50.42,   51.26,   52.10,   52.94 /
      DATA (CHI025(I), I = 76, 150) /
     &         53.78,    54.62,   55.46,   56.30,   57.1532,
     &         57.80,    58.84,   59.69,   60.54,   61.39,
     &         62.24,    63.09,   63.94,   64.79,   65.6466,
     &         66.50,    67.35,   68.21,   69.07,   69.92,
     &         70.78,    71.64,   72.50,   73.36,   74.2219,
     &         75.08,    75.94,   76.80,   77.67,   78.53,
     &         79.40,    80.27,   81.13,   82.00,   82.87,
     &         83.73,    84.60,   85.47,   86.34,   87.21,
     &         88.08,    88.95,   89.83,   90.70,   91.57,
     &         92.45,    93.32,   94.19,   95.07,   95.94,
     &         96.82,    97.70,   98.57,   99.45,   100.33,
     &         101.21,   102.09,  102.97,  103.85,  104.73,
     &         105.61,   106.49,  107.37,  108.25,  109.14,
     &         110.02,   110.90,  111.79,  112.67,  113.56,
     &         114.44,   115.33,  116.21,  117.10,  117.98 /
      DATA (CHI975(I), I = 1, 75) /
     &         5.02389,  7.37776, 9.34840, 11.1433, 12.8325,
     &         14.4494,  16.0128, 17.5346, 19.0228, 20.4831,
     &         21.9200,  23.3367, 24.7356, 26.1190, 27.4884,
     &         28.8454,  30.1910, 31.5264, 32.8523, 34.1696,
     &         35.4789,  36.7807, 38.0757, 39.3641, 40.6465,
     &         41.9232,  43.1944, 44.4607, 45.7222, 46.9792,
     &         48.23,    49.48,   50.72,   51.96,   53.20,
     &         54.44,    55.67,   56.89,   58.12,   59.3417,
     &         60.56,    61.77,   62.99,   64.20,   65.41,
     &         66.62,    67.82,   69.02,   70.22,   71.4202,
     &         72.61,    73.81,   75.00,   76.19,   77.38,
     &         78.57,    79.75,   80.93,   82.12,   83.2976,
     &         84.48,    85.65,   86.83,   88.00,   89.18,
     &         90.35,    91.52,   92.69,   93.86,   95.0231,
     &         96.19,    97.35,   98.52,   99.68,   100.84 /
      DATA (CHI975(I), I = 76, 150) /
     &         102.00,   103.16,  104.31,  105.47,  106.629,
     &         107.78,   108.94,  110.09,  111.24,  112.39,
     &         113.54,   114.69,  115.84,  116.99,  118.136,
     &         119.28,   120.43,  121.57,  122.72,  123.86,
     &         125.00,   126.14,  127.28,  128.42,  129.561,
     &         130.70,   131.84,  132.98,  134.11,  135.25,
     &         136.38,   137.52,  138.65,  139.79,  140.92,
     &         142.05,   143.18,  144.31,  145.44,  146.57,
     &         147.70,   148.83,  149.96,  151.09,  152.21,
     &         153.34,   154.47,  155.59,  156.72,  157.84,
     &         158.97,   160.09,  161.21,  162.33,  163.46,
```

```
     &              164.58,    165.70,    166.82,    167.94,    169.06,
     &              170.18,    171.30,    172.41,    173.53,    174.65,
     &              175.77,    176.88,    178.00,    179.12,    180.23,
     &              181.35,    182.46,    183.58,    184.69,    185.80 /

C  VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:
C
C     1 - 30, 40, 50, 60, 70, 80, 90, 100 — Theil, pp. 718-719
C
C     31-39, 41-49, 51-59, 61-69, 71-79, 81-89, 91-99, 101-150
C        — CALCULATED USING CUBE RULE APPROXIMATION

      DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447,
     &             2.365, 2.306, 2.262, 2.228, 2.201, 2.179,
     &             2.160, 2.145, 2.131, 2.120, 2.110, 2.101,
     &             2.093, 2.086, 2.080, 2.074, 2.069, 2.064,
     &             2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /

C                        LIST OF VARIABLES
C
C  ARG         INTERMEDIATE CALCULATION VARIABLE
C  CHI025()    TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHI975()    TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHITAB      MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  I           CONTROLS LOOP FOR CHI025() AND CHI975()
C  IOUT        OUTPUT DUMP CONTROLLER
C  IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                FOR EACH REGION
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
C                MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL
C                DATA SETS IN A REGION (Number of Points Per Region)
C  NUM         EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  SUHAT       EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SUHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT       EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SWHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SX          EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
C  SX2()       1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                (X = Ln S)
C  T           VALUE OF T025() USED IN CALCULATIONS
C  T025()      TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
C  TTAB        MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C  INITIALIZE IZERO, JZERO AND MCHAT

      DO 50 L = 1, MAXREG
         IZERO(1,L) = 0.0
         IZERO(2,L) = 0.0
         JZERO(1,L) = 0.0
         JZERO(2,L) = 0.0
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
   50 CONTINUE

C  CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

      DO 75 L = 1, NUMREG
         IF (NPPR(L) .GT. CHITAB) THEN
            WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON DEGREES OF FREEDOM ',
```

```
     &                  'IN CHI-SQUARE TABLE, IN REGION ', L
            CALL TRMNAT
         ENDIF
   75 CONTINUE

C  ASSIGN VALUES TO NUM, T, SWHAT, SUHAT AND THEN CALCULATE
C  CONFIDENCE INTERVALS FOR EACH REGION

      DO 100 L = 1, NUMREG

         NUM = NPPR(L)

         IF (NUM .LT. 31) THEN
            T = T025(NUM)
         ELSE
            T = T025(NUM)
         ENDIF

         SWHAT = SWHAT2(L) ** 0.5
         SUHAT = SUHAT2(L) ** 0.5
         SX = (NUM * SX2(L)) ** 0.5

C        CALCULATE ESTIMATED VALUES OF M AND C

         ARG = T * SWHAT / SX
         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SUHAT

C        CALCULATE CONFIDENCE INTERVALS

         IZERO(1,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI975(NUM)) ** 0.5
         IZERO(2,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI025(NUM)) ** 0.5
         JZERO(1,L) = MCHAT(1,L) - ARG
         JZERO(2,L) = MCHAT(1,L) + ARG


         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L =', L, ' NPPR =', NPPR(L), ' NUM =', NUM
            WRITE(8,*) 'SWHAT2 =', SWHAT2(L), ' SWHAT =', SWHAT
            WRITE(8,*) 'SUHAT2 =', SUHAT2(L), ' SUHAT =', SUHAT
            WRITE(8,*) 'SX2 =', SX2(L), ' SX =', SX
            WRITE(8,*) 'CHI025 =', CHI025(NUM), ' CHI975 =', CHI975(NUM)
            WRITE(8,*) 'T =', T, ' DD =', DD(L), ' ARG =', ARG
            WRITE(8,*) 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =',
     &                  IZERO(2,L)
            WRITE(8,*) 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =',
     &                  JZERO(2,L)
            WRITE(8,*) 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =',
     &                  MCHAT(2,L)
         ENDIF
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE GTPVAR CALCULATES THE EXTENT OF DEPARTURE FROM THE MULTIPLE
C  HEAT MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)

C  INPUTS:   NSETS, NP, NUMREG, LAMN, MCHAT
C  OUTPUTS:  PVAR

C     IMPLICIT NONE
```

```
      INTEGER MAXREG, MAXSET

      PARAMETER (MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG),
     &        NUMREG, TOTAL

      REAL    LAMN, MCHAT(2, MAXREG), PSIG2(MAXREG), PVAR, SUM


C                     LIST OF VARIABLES
C
C   IOUT         OUTPUT DUMP CONTROLLER
C   J            CONTROLS DO LOOP FOR EACH DATA SET
C   L            CONTROLS DO LOOP FOR EACH REGION
C   LAMN         LAMBDA-N — RATIO OF Var (Ln N given S) / (m**2 C**2),
C                  CONSTANT OVER REGIONS AND COMPONENTS
C   MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C   MAXSET       MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C   MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                  FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
C                  MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
C                  MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   NP()         2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                  SET IN EACH REGION
C   NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUM()        EQUAL TO Nj-1 FOR EACH REGION WHERE Nj IS THE SUM OF THE
C                  NUMBER OF POINTS IN EACH DATA SET
C   NUMREG       NUMBER OF REGIONS OF INTEREST
C   PSIG2()      1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
C                  VARIATION IN EACH REGION
C   PVAR         THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C                  CURVE WARRANTED BY THE AVAILABLE INFORMATION
C   SUM          WEIGHTED SUM OF THE PSIG2s — USED TO CALCULATE A WEIGHTED
C                  AVERAGE
C   TOTAL        SUM OF NUM() OVER ALL REGIONS

C     INITIALIZE VARIABLES

      SUM = 0.0
      TOTAL = 0.0

      DO 50 L = 1, MAXREG
         PSIG2(L) = 0.0
         NUM(L) = 0
   50 CONTINUE

      DO 100 L = 1, NUMREG
         DO 150 J = 0, NSETS
            NUM(L) = NUM(L) + NP(J,L)
  150    CONTINUE
         NUM(L) = NUM(L) - 1
         TOTAL = TOTAL + NUM(L)
  100 CONTINUE

      DO 200 L = 1, NUMREG
         PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
         SUM = SUM + PSIG2(L) * NUM(L)
  200 CONTINUE

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'LAMN = ', LAMN
         DO 300 L = 1, NUMREG
            WRITE(8,*) 'L = ', L, ' NUM = ', NUM(L)
            WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
  300    CONTINUE
         WRITE(8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
      ENDIF

      PVAR = SUM / FLOAT (TOTAL)
```

```
      RETURN
      END

C***********************************************************************

C  SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
C  M AND Co WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C  TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
     &                   MCHAT, RANGEM)

C  INPUTS:   NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C  OUTPUTS:  RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
     &        MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

C
C                          LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  JZERO()      2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LOWER        LOWER BOUND OF INTERSECTION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()         2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                 REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                 - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                 BOUND
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
C                 FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MC() FOR EACH REGION
C  MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  UPPER        UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         RANGEM(1,L) = 0.0
         RANGEM(2,L) = 0.0
   50 CONTINUE

C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST
```

```
      DO 100 L = 1, NUMREG

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
         ENDIF

         IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS NO EXOGENOUS INFORMATION
C           ASSUME RANGE TO BE Jo

            RANGEM(1,L) = JZERO(1,L)
            RANGEM(2,L) = JZERO(2,L)

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' JZERO(1,L) = ', JZERO(1,L)
               WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN

C           NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE
C           TO Co, ADJUST THE LOWER BOUND OF Jo ACCORDINGLY

            LOWER = AMAX1(JZERO(1,L), MC(1,L))
            UPPER = JZERO(2,L)
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN

C           THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE
C           CORRESPONDING TO THE Co CONSTRAINT, ADJUST Jo ACCORDINGLY

            LOWER = AMAX1(JZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF (MPNT(L) .EQ. 1) THEN

C           THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER
C           INFORMATION:  ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
```

```
            RANGEM(1,L) = MZERO(1,L)
            RANGEM(2,L) = 0.0

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS A PRIOR RANGE ON M, BUT NO Co CONSTRAINT
C           USE INTERSECTION BETWEEN Jo AND Mo

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mo'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN

C           THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C           CONSTRAINT, INTERSECT Jo AND Mo, ADJUSTING THE LOWER BOUND
C           BY Mc ACCORDINGLY

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                    'AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANEGM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN

C           THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO Co CONSTRAINT
C           INTERSECT THESE TWO RANGES WITH Jo

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                    'AND Mc'
               CALL TRMNAT
            ELSE
```

```
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSE

            WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
            CALL TRMNAT

         ENDIF

C        RESTRICT RANGE TO BE NON-NEGATIVE

         RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

         IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &          .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END

C***********************************************************************


C  SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C  WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88     FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  INPUTS:   RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C  OUTPUTS:  RANGEM, MCHAT, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG
```

```
      REAL      MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)

C                  LIST OF VARIABLES
C
C   IOUT          OUTPUT DUMP CONTROLLER
C   L             CONTROLS DO LOOP FOR EACH REGION
C   LL            EQUAL TO NUMREG FOR A SET OF CALCULATIONS
C   MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C   MCHAT()       2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C                   C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                   MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND
C                   MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   MPNT()        1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                   MZERO() FOR EACH REGION
C   MZERO()       2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                   EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                   IS UPPER BOUND
C   NNODAT        Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NUMREG        NUMBER OF REGIONS OF INTEREST
C   RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                   FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                   RANGEM(2,L) IS THE UPPER BOUND

      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
            ENDIF

C           SPECIFY E(M) OF POSTERIOR FOR SAKE OF
C           CALCULATIONS IN SUBROUTINE EXPCTD

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE CONCAV ADJUSTS THE UPPER BOUNDS OF THE POSTERIOR CREDIBILITY
C  RANGES ON M TO BE CONSISTENT WITH CONCAVITY CONSTRAINTS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  2FEB88      FORMAT/COMMENTS:  15SEP89
```

```
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONCAV (NUMREG, RANGEM)

C  INPUTS:    NUMREG, RANGEM
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    RANGEM(2, MAXREG), TESTM


C                             LIST OF VARIABLES
C
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  RANGEM()   2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C               FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C               RANGEM(2,L) IS THE UPPER BOUND
C  TESTM     UPPER BOUND OF RANGE ON M IN REGION L-1 - USED DURING
C              CONCAVITY ADJUSTMENT


C     ADJUST RANGE TO INSURE CONCAVITY

      DO 100 L = NUMREG, 2, -1

         IF (RANGEM(2,L-1) .EQ. 0.0) THEN
C           RANGE IS A POINT IN REGION L-1
            IF (RANGEM(1,L-1) .GT. AMAX1(RANGEM(1,L),RANGEM(2,L))) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &            ' IS INCONSISTENT WITH POINT POSTERIOR IN REGION ',L-1
               CALL TRMNAT
            ENDIF
         ELSE
C           RANGE IS AN INTERVAL IN REGION L-1
            TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
            IF (TESTM .LT. RANGEM(1,L-1)) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &            ' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
     &            'REGION ', L-1
               CALL TRMNAT
            ELSE
               RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
            ENDIF
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,L-1) =', RANGEM(1,L-1),
     &                 ' RANGEM(2,L-1) =', RANGEM(2,L-1)
            WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                 ' RANGEM(2,L) =', RANGEM(2,L)
            WRITE(8,*) 'TESTM =', TESTM, ' L = ', L
         ENDIF

  100 CONTINUE

      RETURN
      END
```

```
C***********************************************************************

C  SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS
C  BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR Co
C  PROGRAMMER:  L. NEWLIN
C       DATE:   CODE:  5OCT87     COMMENTS:  1DEC87
C    VERSION:   MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C  INPUTS:    NUMREG, RANGEM
C  IOUTPUT:   MEDM

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG

      REAL     LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)


C                    LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LOWERM       LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
C                 TO BE USED IN MEDIAN CALCULATION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEDM()       1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND

C     INITIALIZE ARRAY MEDM

      DO 50 L = 1, MAXREG
         MEDM(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATIONS FOR EACH REGION

      DO 100 L = 1, NUMREG

         IF (RANGEM(2,L) .EQ. 0.0) THEN

C           RANGE IS A POINT

            MEDM(L) = RANGEM(1,L)

         ELSEIF (L .EQ. 1) THEN

C           WE ARE IN REGION ONE - NOT AFFECTED BY OTHER REGIONS
C           - MEDIAN WILL JUST BE AVERAGE OF RANGEM VALUES

            MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0

         ELSE

C           MUST TAKE MEDIAN OF REGION L-1 INTO ACCOUNT

            LOWERM = AMAX1(RANGEM(1,L), MEDM(L-1))
            MEDM(L) = (LOWERM + RANGEM(2,L)) / 2.0
```

```
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
            WRITE(8,*) 'LOWERM = ', LOWERM, ' MEDM(L) = ', MEDM(L)
         ENDIF

  100 CONTINUE

      RETURN
      END


C*************************************************************************


C  SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES OF THE S/N
C  CURVE PARAMETERS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  13FEB89      FORMAT/COMMENTS:  15SEP89
C     VERSION:  MATCHR V8.3, V8.4, V8.5    MATGRM V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
     &                   ZROREG, NBND, BIGK1, BZHAT)

C  INPUTS:  NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
C  OUTPUTS:  BIGK1, BZHAT
C  SUBPROGRAMS:  TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG

      REAL     BIGK(0:MAXREG), BIGK1, BZHAT, FACTR, KHAT, MEANZ,
     &         MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
     &         NF(MAXDAT, MAXREG), SBND(0:MAXREG), STR(MAXDAT, MAXREG),
     &         SZ2, SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)


C                    LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  BIGK1      EQUAL TO BIGK(1)
C  BZHAT      E(BETAo)
C  FACTR      A SCALE FACTOR = PHI * KRATIO * Z
C  IOUT       OUTPUT DUMP CONTROLLER
C  KHAT       E(k)
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MEDM()     1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C  MM()       1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NCOMPS     Number of COMPonents - 1 FOR STRESS AND STRAIN WHEN DECOMPOSED
C               DATA UNAVAILABLE - 2 FOR DECOMPOSED STRAIN DATA
C  NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
```

```
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C                  DATA SET
C   NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR
C                  THE SPECIFIC MATERIAL S/N DATA SET
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   SBND()      1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C                 CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
C                 CONTAINED IN NBND()
C   STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C                 DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SZ2         SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C   SZERO       STRESS TENSILE TEST POINT, So
C   TRBIGK()    1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
C                 TRBIGK(i) = BIGK(i)
C   ZROREG      ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C   ZZ()        1-D ARRAY CONTAINING TRANSFORMED S-N DATA, Z = F(STR,NF,NBND,MM)

C  INITIALIZE VARIABLES

      DO 50 L = 0, MAXREG
         MM(L) = 0.0
   50 CONTINUE

C  CREATE MM() ARRAY FROM MEDM() ARRAY

      DO 100 L = 1, NUMREG
         MM(L) = MEDM(L)
  100 CONTINUE

C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

      CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C  CALCULATE BETAo AND k

      CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

C  CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

      CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

      BIGK1 = BIGK(1)

C  CALCULATE BOUNDARIES OF STRESS REGIONS

      CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C  CALCULATE Ko AND Mo FOR THE NO DATA REGION TO THE LEFT IF REQUIRED

      DO 150 L = ZROREG, NUMREG
         TRBIGK(L) = BIGK(L)
  150 CONTINUE

      IF (ZROREG .EQ. 0) THEN
         FACTR = 1.0
         CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK,
     &              FACTR, NUMREG)
      ENDIF

C  WRITE RESULTS TO FILE

      IF (NCOMPS .EQ. 1) THEN

         WRITE(7,900) NUMREG, BZHAT, KHAT
         IF (IOUT .EQ. 10) WRITE(8,900) NUMREG, BZHAT, KHAT

         DO 200 L = ZROREG, NUMREG
            WRITE(7,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
```

```
            IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
     &                                     NBND(L), SBND(L)
 200     CONTINUE

         WRITE(7,920)

      ELSE

         WRITE(7,930) MM(1), BIGK(1), KHAT

      ENDIF

C  FORMAT STATEMENTS

 900  FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',//,2X,
     &       'NUMBER OF REGIONS:',I4,5X,'E(BETAo) =',F8.4,5X,'E(k) =',
     &       F8.4,//,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',7X,
     &       'STRESS BOUND',/)

 910  FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E9.3,9X,E11.5)

 920  FORMAT(///)

 930  FORMAT(//,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
     &       //,11X,'m',14X,'K',13X,'E(k)',
     &       //,7X,F8.5,5X,E12.5,6X,F7.4,/)

      RETURN
      END

C***********************************************************************

C  SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C  MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA,
     &                  MO, SIGMA2, MCHAT, MU, SIG)

C   INPUTS:  NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2
C  OUTPUTS:  MCHAT, MU, SIG

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG, NPPR(MAXREG)

      REAL     ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG),
     &         MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG),
     &         SUHAT2(MAXREG), SUMX2, SWHAT2(MAXREG), SX2(MAXREG)

C                LIST OF VARIABLES
C
C  ARG          INTERMEDIATE CALCULATION VARIABLE
C  DD()         1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()      1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                 SIG() CALCULATION
C  IOUT         OUTPUT DUMP CONTROLLER
C  L            CONTROLS DO LOOP FOR EACH REGION
C  MAXREG       MAXIMUM NUMBER OF REGION ALLOWED
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR
```

```
C                    EACH REGION, BASED ON MATERIALS DATA ONLY - MCHAT(1,L) =
C                    - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT,
C                    THE ESTIMATE FOR C
C   MO()          1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                    MEAN FOR EACH REGION
C   MU()          1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                    DISTRIBUTION MEAN FOR EACH REGION
C   NPPR()        1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL
C                    DATA SETS IN A REGION (Number of Points Per Region)
C   NUMREG        NUMBER OF REGIONS OF INTEREST
C   SIG()         1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                    DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C   SIGMA2()      1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                    VARIANCE FOR EACH REGION
C   SUHAT2()      1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                    REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SUMX2         EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
C   SWHAT2()      1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                    REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SX2()         1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                    (X = Ln S)

C     INITALIZE ARRAYS

      DO 50 L = 1, MAXREG
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
         MU(L) = 0.0
         SIG(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATION FOR EACH REGION

      DO 100 L = 1 , NUMREG

         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SQRT (SUHAT2(L))
         SUMX2 = NPPR(L) * SX2(L)
         ARG = SUMX2 + DELTA(L)

         IF (DELTA(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
C           USE THE ESTIMATE OF M
            MU(L) = MCHAT(1,L)
         ELSE
C           UPDATE THE ESTIMATE OF M WITH Mo USING DELTA
            MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
         ENDIF

         IF (SIGMA2(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
C           USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
            SIG(L) = SQRT (SWHAT2(L) / ARG)
         ELSE
            SIG(L) = SQRT (SIGMA2(L) / ARG)
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ',
     &                  MCHAT(1,L)
            WRITE(8,*) 'SUHAT2 = ', SUHAT2(L),  ' MCHAT2 = ',
     &                  MCHAT(2,L)
            WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
     &                 ' SUMX2 = ', SUMX2
            WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
            WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
            WRITE(8,*) 'SWHAT2 = ', SWHAT2(L), ' SIGMA2 = ', SIGMA2(L),
     &                 ' SIG = ', SIG(L)
         ENDIF

  100 CONTINUE

      RETURN
```

```
      END

C***********************************************************************

C   SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND Co TO
C   OBTAIN POSTERIOR RANGES ON M FOR EACH REGION
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  10FEB88      FORMAT/COMMENTS:  12AUG91
C      VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C   INPUTS:    NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C   OUTPUTS:   RANGEM
C   SUBPROGRAMS:   TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL     LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
     &         RANGEM(2, MAXREG), UPPER

C                          LIST OF VARIABLES
C
C   IOUT       OUTPUT DUMP CONTROLLER
C   L          CONTROLS DO LOOP FOR EACH REGION
C   LOWER      LOWER BOUND OF INTERSECTION
C   MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C   MC()       2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                BOUND
C   MCHAT()    2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
C                FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   MCPNT()    1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MC() FOR EACH REGION
C   MPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C   MZERO()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
C   NUMREG     NUMBER OF REGIONS OF INTEREST
C   RANGEM()   2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C   UPPER      UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         RANGEM(1,L) = 0.0
         RANGEM(2,L) = 0.0
   50 CONTINUE

C   PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
```

```
      ENDIF

      IF (MPNT(L) .EQ. 1) THEN

C        THERE IS A POINT PRIOR ON M - THIS OVERRIDES ALL OTHER
C        INFORMATION:  ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR

         RANGEM(1,L) = MZERO(1,L)
         RANGEM(2,L) = 0.0

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN

C        THERE IS A PRIOR RANGE ON M, BUT NO Co CONSTRAINT USE Mo

            RANGEM(1,L) = MZERO(1,L)
            RANGEM(2,L) = MZERO(2,L)

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN

C        THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C        CONSTRAINT ADJUST THE LOWER BOUND OF Mo BY Mc

         LOWER = AMAX1(MZERO(1,L), MC(1,L))
         UPPER = MZERO(2,L)
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANEGM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN

C        THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO Co CONSTRAINT
C        INTERSECT THESE TWO RANGES

         LOWER = AMAX1(MZERO(1,L), MC(1,L))
         UPPER = AMIN1(MZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
```

```
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSE

            WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
            CALL TRMNAT

         ENDIF

C        RESTRICT RANGE TO BE NON-NEGATIVE

         RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

         IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &        .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END

C************************************************************************

C  SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C  DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  10FEB88     FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
     &                   MZERO, MPNT, MO, SIGMA2)

C  INPUTS:    RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
C             MO, SIGMA2
C  OUTPUTS:   RANGEM, MCHAT, MU, SIG, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

      REAL    MCHAT(2, MAXREG), MO(MAXREG), MU(MAXREG),
     &        MZERO(2, MAXREG), RANGEM(2, MAXREG), SIG(MAXREG),
     &        SIGMA2(MAXREG)


C                  LIST OF VARIABLES
C
C  IOUT           OUTPUT DUMP CONTROLLER
C  L              CONTROLS DO LOOP FOR EACH REGION
```

```
C  LL              EQUAL TO NUMREG FOR A SET OF CALCULATIONS
C  MAXREG          MAXIMUM NUMBER OF REGIONS ALLOWED
C  MCHAT()         2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C                    C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                    MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND
C                    MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MO()            1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                    MEAN FOR EACH REGION
C  MPNT()          1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                    MZERO() FOR EACH REGION
C  MU()            1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                    DISTRIBUTION MEAN FOR EACH REGION
C  MZERO()         2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                    EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                    IS UPPER BOUND
C  NNODAT          Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NUMREG          NUMBER OF REGIONS OF INTEREST
C  RANGEM()        2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                    FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                    RANGEM(2,L) IS THE UPPER BOUND
C  SIG()           1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                    DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C  SIGMA2()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                    VARIANCE FOR EACH REGION

      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            MU(LL) = MO(LL)
            SIG(LL) = SQRT(SIGMA2(LL))
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
               WRITE(8,*) 'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
               WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
     &                    SIGMA2(LL)
            ENDIF

C           SPECIFY E(M) OF POSTERIOR FOR SAKE OF
C           CALCULATIONS IN SUBROUTINE EXPCTD

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
               MU(LL) = RANGEM(1,LL)
               SIG(LL) = 0.0
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL),
     &         ' MU = ', MU(LL), ' SIG = ', SIG(LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE

      RETURN
      END


C***********************************************************************
```

```
C   SUBROUTINE EXPB CALCULATES THE EXPECTED VALUES OF THE S/N
C   CURVE PARAMETERS FOR THE BOOTSTRAP IMPLEMENTATION
C   PROGRAMMER:  L. NEWLIN
C         DATE:  11OCT90     COMMENTS:  15JAN92
C      VERSION:  MATCHR VB1.1, VB1.2, VB1.3
C                MATGRM VB1, VB1.1
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.

      SUBROUTINE EXPB (MM, BIGK, SZERO, NUMREG, ZROREG, NBND)

C   INPUTS:  MM, BIGK, SZERO, NUMREG, ZROREG, NBND
C   SUBPROGRAMS:  FINDSB, KOMO

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG, ZROREG

      REAL    BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     &        SBND(0:MAXREG), SZERO, TRBIGK(0:MAXREG)


C                    LIST OF VARIABLES
C
C   BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C                EACH REGION
C   FACTR      A SCALE FACTOR = PHI * KRATIO * Z
C   IOUT       OUTPUT DUMP CONTROLLER
C   L          CONTROLS DO LOOP FOR EACH REGION
C   MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C   MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C   MM()       1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C   NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C   NUMREG     NUMBER OF REGIONS OF INTEREST
C   SBND()     1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C                CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
C                CONTAINED IN NBND()
C   SZERO      STRESS TENSILE TEST POINT, So
C   TRBIGK()   1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
C                TRBIGK(i) = BIGK(i)
C   ZROREG     ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION


C   CALCULATE BOUNDARIES OF STRESS REGIONS

      CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C   CALCULATE Ko AND Mo FOR THE NO DATA REGION TO THE LEFT IF REQUIRED

      DO 150 L = ZROREG, NUMREG
         TRBIGK(L) = BIGK(L)
  150 CONTINUE

      IF (ZROREG .EQ. 0) THEN
         FACTR = 1.0
         CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK,
     &                   FACTR, NUMREG)
      ENDIF

C   WRITE RESULTS TO FILE
```

```
         WRITE(7,900) NUMREG
         IF (IOUT .EQ. 10) WRITE(8,900) NUMREG

         DO 200 L = ZROREG, NUMREG
            WRITE(7,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
            IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
     &                                     NBND(L), SBND(L)
  200    CONTINUE

         WRITE(7,920)

C  FORMAT STATEMENTS

  900 FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
     &       //,2X,'NUMBER OF REGIONS:',I4,
     &       //,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',7X,
     &       'STRESS BOUND',/)

  910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E9.3,9X,E11.5)

  920 FORMAT(///)

      RETURN
      END


C*************************************************************************


C  SUBROUTINE PEB CONTROLS THE CALCULATIONS FOR THE Parameter Estimation
C  MODEL PORTION OF THE MATERIALS CHARACTERIZATION Bootstrap MODEL
C  PROGRAMMER:  L. NEWLIN
C        DATE:  13NOV90      FORMAT/COMMENTS:  15JAN92
C     VERSION:  MATCHR VB1.2, VB1.3
C               MATGRM VB1, VB1.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.


      SUBROUTINE PEB (NPTS, NUMREG, ZROREG, RAND, NBND, STR, MEDM,
     &                MEDK, RESID, BIGK, BZERO, MM, SBND)

C        INPUTS:  NPTS, NUMREG, ZROREG, RAND, NBND, STR, MEDM, MEDK, RESID
C       OUTPUTS:  BIGK, BZERO, MM, SBND
C  SUBPROGRAMS:  PICRES, MREGR, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, ZROREG

      REAL     BIGK(0:MAXREG), BZERO, K, MEANZ, MEDK(0:MAXREG),
     &         MEDM(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
     &         RESID(MAXDAT), RESNF(MAXDAT, MAXREG), SBND(0:MAXREG),
     &         STR(MAXDAT, MAXREG), SZ2, ZZ(MAXDAT)

      DOUBLE PRECISION RAND

C                      LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C              EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C  IOUT      OUTPUT DUMP CONTROLLER
```

```
C  K         VALUE OF k — PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C              DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT    MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ     SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MEDK()    1-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION
C              (BOOTSTRAP OPTION)
C  MEDM()    1-D ARRAY CONTAINING THE MEAN VALUES M FOR EACH REGION
C              (BOOTSTRAP OPTION)
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NP        TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()    1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C              SPECIFIC MATERIAL S/N DATA SET
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  RAND      RANDOM NUMBER SEED
C  RESID()   1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH
C              POINT IN THE SPECIFIC MATERIAL S/N DATA SET
C  RESNF()   2-D ARRAY CONTAINING NF() (CYCLES TO FAILURE) FOR THE SPECIFIC
C              MATERIAL S/N DATA SET BROKEN INTO REGIONS BASED ON THE
C              RANDOMLY SELECTED RESIDUALS
C  SBND()    1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C              REGION CONTAINED IN NBND()
C  STR()     2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C              DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SZ2       SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  ZROREG    ZeRO REGion — VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C  ZZ()      1-D ARRAY CONTAINING TRANSFORMED S/N DATA, Z = F(STR,NF,NBND,MM)

C  OBTAIN THE VALUES OF THE RANDOMLY SELECTED RESIDUALS FOR EACH DATA POINT

      CALL PICRES (RAND, MEDM, MEDK, RESID, NPTS, STR, RESNF)

C  BOOTSTRAPPING M IS DESIRED

      CALL MREGR (NUMREG, NPTS, STR, RESNF, MM)

C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

      CALL TRNSFM (NPTS, STR, RESNF, NUMREG, MM, NBND, NP, ZZ)

C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C  CALCULATE THE VALUES FOR k AND BETAo FROM THE SAMPLE MEAN
C  AND VARIANCE

      CALL KBETA (MEANZ, SZ2, K, BZERO)

C  CALCULATE THE VALUE OF K FOR EACH REGION WHERE A = K ** M

      CALL FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C  CALCULATE STRESS TIE-POINTS

      CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C  WRITE RESULTS TO FILE

C       WRITE(7,900) NUMREG, BZERO

C       DO 200 L = ZROREG, NUMREG
C          WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
C  200 CONTINUE

C       WRITE(7,920)

C  FORMAT STATEMENTS
```

```
  900 FORMAT(///,2X,'SELECTED VALUES OF S/N CURVE PARAMETERS',
     &        //,2X,'NUMBER OF REGIONS: ',I4,5X,'BETAo = ',F8.4,
     &        //,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',5X,
     &        'STRESS BOUND',/)

  910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E9.3,6X,E11.5)

  920 FORMAT(///)

      RETURN
      END


C************************************************************************


C***********************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C***********************************************************************

      SUBROUTINE RANDOM (FRAC, RAND)
C     IMPLICIT NONE
      COMMON  IOUT
      INTEGER IOUT
      REAL    FRAC
      DOUBLE PRECISION  RANA, RANC, RAND, RANDIV, RANM, RANSUB,
     &          RANT, RANX


C                 LIST OF VARIABLES
C
C   FRAC       UNIFORM (0,1) RANDOM VARIATE
C   IOUT       OUTPUT DUMP CONTROLLER
C   RANA       CONSTANT FOR LCG
C   RANC       CONSTANT FOR LCG
C   RAND       RANDOM NUMBER SEED
C   RANDIV     INTERNAL CALCULATION
C   RANM       CONSTANT FOR LCG
C   RANSUB     INTERNAL CALCULATION
C   RANT       INTERNAL CALCULATION
C   RANX       INTERNAL CALCULATION


C           USING LCG RANDOM # GENERATOR

      RANA = 671093.0
      RANC = 7090885.0
      RANM = 33554432.0

   10 RANX = RANA * RAND + RANC
      RANDIV = RANX / RANM
      RANT = DINT(RANDIV)
      RANSUB = RANT * RANM
      RAND = RANX - RANSUB
      FRAC = SNGL(RAND / RANM)

      IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
      IF (IOUT .EQ. 2) WRITE(8,*)'RANX =', RANX, ' RANDIV =', RANDIV,
```

```
     &    ' RANT =', RANT, ' RANSUB =', RANSUB, ' RAND =', RAND,
     &    ' FRAC =', FRAC

      RETURN
      END

C     NOTES:  IOUT=2 DUMPS TO SCREEN


C**********************************************************************


C**********************************************************************
C  SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C  WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  3FEB88
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
C
C  The random variates are generated using the "Direct Method"
C  Abramowitz, M., and Stegun, I. A., editors, Handbook of
C    Mathematical Functions, National Bureau of Standards, Applied
C    Mathematics Series 55, Issued June 1964, Ninth Printing, November
C    1970 with corrections, pg. 953.
C**********************************************************************

      SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C     SUBPROGRAM:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2

      PARAMETER (PI = 3.1415926536)

      INTEGER IOUT


C             LIST OF VARIABLES
C
C  FRAC     UNIFORM(0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  MU       MEAN OF NORMAL DISTRIBUTION
C  RAND     RANDOM NUMBER SEED
C  SIGMA    STANDARD DEVIATION OF NORMAL DISTRIBUTION
C  X        NORMAL RANDOM VARIATE
C  U1       UNIFORM RANDOM NUMBER U(0,1)
C  U2       UNIFORM RANDOM NUMBER U(0,1)
C  Z1       NORMAL RANDOM NUMBER ON N(0,1)
C  Z2       NORMAL RANDOM NUMBER ON N(0,1)


      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'RAND =', RAND, ' MU =', MU, ' SIGMA =', SIGMA

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

      CALL RANDOM (FRAC, RAND)
      U2 = FRAC
      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'U1 =', U1,' U2 =', U2

      Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
      Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)

      X = SIGMA * Z1 + MU
```

```
      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'Z1 =', Z1, ' Z2 =', Z2, ' X =', X

      RETURN
      END


C***********************************************************************


C  SUBROUTINE PICRES PERFORMS THE RESIDUAL SELECTION AND THEN CALCULATES
C  THE NEW S/N PAIRS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10OCT90
C     VERSION:  MATCHR VB1.1, VB1.2, VB1.3
C               MATGRM VB1, VB1.1

      SUBROUTINE PICRES (RAND, MEDM, MEDK, RESID, NPTS, STR, RESNF)

C   INPUTS:  RAND, MEDM, MEDK, RESID, NPTS, STR
C  OUTPUTS:  RESNF

C     IMPLICIT NONE

      INTEGER  MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON   IOUT

      INTEGER  I, INDEX, IOUT, K, L, NPTS(MAXREG)

      REAL     LNK, MEDK(0:MAXREG), MEDM(0:MAXREG), RESID(MAXDAT),
     &         RESNF(MAXDAT, MAXREG), STR(MAXDAT, MAXREG), X

      DOUBLE PRECISION  RAND


C               LIST OF VARIABLES
C
C  I         CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C  INDEX     THE RANDOMLY SELECTED INDEX FOR THE RESIDUAL SELECTION
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         CONTROLS DO LOOP FOR EACH POINT IN EACH REGION
C  L         CONTROLS DO LOOP FOR EACH REGION
C  LNK       EQUAL TO ln(K)
C  MAXDAT    MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED'
C  MEDK()    1-D ARRAY CONTAINING THE MEDIAN K FOR EACH REGION
C  MEDM()    1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C  NPTS()    1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C              (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C  RAND      RANDOM NUMBER SEED
C  RESID()   1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH
C              POINT IN THE SPECIFIC MATERIAL S/N DATA SET
C  RESNF()   2-D ARRAY CONTAINING THE CALCULATED CYCLES TO FAILURE FOR THE
C              SPECIFIC MATERIAL S/N DATA SET AND SELECTED RESIDUALS
C  STR()     2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C              S/N DATA SET BROKEN INTO REGIONS (PSI)
C  X         UNIFORM(0,1) RANDOM VARIATE


      DO 50 L = 1, MAXREG
         DO 75 K = 1, MAXDAT
            RESNF(K,L) = 0.0
   75    CONTINUE
   50 CONTINUE

      LNK = ALOG (MEDK(1))

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'MEDK = ', MEDK(1), ' LNK = ', LNK
         WRITE(8,*) 'NPTS = ', NPTS(1), ' MEDM = ', MEDM(1)
```

```
      ENDIF

      DO 100 I = 1, NPTS(1)
         CALL RANDOM (X, RAND)
         INDEX = INT (X * FLOAT (NPTS(1))) + 1
         RESNF(I,1) = EXP (MEDM(1) * (LNK - ALOG (STR(I,1)))
     &                   + RESID(INDEX))
         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'STR = ', STR(I,1), ' RESNF = ', RESNF(I,1)
            WRITE(8,*) 'X = ', X, ' INDEX = ', INDEX,
     &                 'RESID = ', RESID(INDEX)
         ENDIF
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE MREGR CALCULATES, M, THE MATERIALS SHAPE PARAMETER
C  FOR EACH REGION WHERE Y = LN(NF) AND X = LN(STR)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  13NOV90
C     VERSION:  MATCHR VB1.2, VB1.3
C               MATGRM VB1, VB1.1

      SUBROUTINE MREGR (NUMREG, NPTS, STR, NF, MM)

C  INPUTS:   NUMREG, NPTS, STR, NF
C  OUTPUTS:  MM

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, K, L, NPTS(MAXREG), NUMREG

      REAL    DIFFX(MAXDAT),  DIFFY(MAXDAT), MM(0:MAXREG),
     &        NF(MAXDAT, MAXREG), STR(MAXDAT, MAXREG), MEANX, MEANY,
     &        SX2(MAXREG), SXY(MAXREG)


C                    LIST OF VARIABLES
C
C  DIFFX()    1-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LN(STR(K,L))
C                AND MEANX FOR EACH POINT IN REGION L
C  DIFFY()    1-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LN(NF(K,L))
C                AND MEANY FOR EACH POINT IN REGION L
C  IOUT       OUTPUT DUMP CONTROLLER
C  K          CONTROLS DO LOOP FOR EACH POINT IN A REGION
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT     MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANX      SAMPLE X MEAN FOR POINTS FROM REGION L (X = Ln S)
C  MEANY      SAMPLE Y MEAN FOR POINTS FROM REGION L (Y = Ln N)
C  MM()       1-D ARRAY CONTAINING VALUES OF THE MEAN M FOR EACH REGION
C               (BOOTSTRAP OPTION)
C  NF()       2-D ARRAY CONTAINING LN(RESNF()), ALSO INDEXED FOR REGION
C  NPTS()     1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  STR()      2-D ARRAY CONTAINING LN(STR()), ALSO INDEXED FOR REGION
C  SX2()      1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C              (X = Ln S)
C  SXY()      1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
C               EACH REGION (X = Ln S, Y = Ln N)
```

```
C  INITIALIZE ARRAYS

      DO 50 L = 1, MAXREG
        SX2(L) = 0.0
        SXY(L) = 0.0
        MM(L) = 0.0
   50 CONTINUE

      DO 70 K = 1, MAXDAT
         DIFFY(K) = 0.0
         DIFFX(K) = 0.0
   70 CONTINUE

C  NOW PERFORM CALCULATION OF SX2 AND SXY, FOR EACH REGION

      DO 100 L = 1, NUMREG

C        FIRST CALCULATE SAMPLE X AND Y MEANS IN REGION L
         MEANY = 0.0
         MEANX = 0.0
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' NPTS =', NPTS(L)

         DO 250 K = 1, NPTS(L)
            MEANY = MEANY + ALOG (NF(K,L))
            MEANX = MEANX + ALOG (STR(K,L))
            IF (IOUT .EQ. 10) WRITE(8,*) 'NF =', NF(K,L),
     &                ' STR =', STR(K,L)
  250    CONTINUE

         MEANY = MEANY/FLOAT(NPTS(L))
         MEANX = MEANX/FLOAT(NPTS(L))
         IF (IOUT .EQ. 10) WRITE(8,*) 'MEANY =', MEANY,
     &     ' MEANX =', MEANX

C        NOW CALCULATE SAMPLE VARIANCES, SX2 AND SXY,
C        OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
C        DATA POINT IN REGION L

         DO 300 K = 1, NPTS(L)
            DIFFY(K) = ALOG (NF(K,L)) - MEANY
            DIFFX(K) = ALOG (STR(K,L)) - MEANX
            SX2(L) = SX2(L) + DIFFX(K) ** 2
            SXY(L) = SXY(L) + DIFFX(K) * DIFFY(K)
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'K =', K, ' DIFFY(K) =', DIFFY(K),
     &                    ' DIFFX(K) =', DIFFX(K)
               WRITE(8,*) 'SX2(L) =', SX2(L), ' SXY(L) =', SXY(L)
            ENDIF
  300    CONTINUE

         IF (SXY(L) .GE. 0.0) THEN
C           LIFE WILL INCREASE WITH INCREASING STRESS - INVALID FOR
C           OUR MODEL
            WRITE(8,*) 'ERROR: SXY >= 0 IN REGION', L
            CALL TRMNAT
         ENDIF

C        NOW CALCULATE THE M FOR REGION L

         MM(L) = - SXY(L) / SX2(L)

         IF (IOUT .EQ. 10) WRITE(8,*) 'SX2(L) =', SX2(L), ' SXY(L) =',
     &                                SXY(L), ' MM(L) =', MM(L)

  100 CONTINUE

      RETURN
      END


C**************************************************************************
```

```
C  SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM
C  THE S/N DATA INTO THE VARIABLE Z = Ln(X)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C  INPUTS:   NPTS, STR, NF, NUMREG, MM, NBND
C  OUTPUTS:  NP, ZZ

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON IOUT

      INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG

      REAL     MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &         STR(MAXDAT, MAXREG), ZZ(MAXDAT)


C                    LIST OF VARIABLES
C
C  I          CONTROLS DO LOOP FOR EACH DATA POINT
C  IOUT       OUTPUT DUMP CONTROLLER
C  K          CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LL         CONTROLS INNER DO LOOP FOR EACH REGION
C  MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
C  MML        EQUAL TO MM(L) FOR A SET OF CALCULATIONS
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C               SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()     1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C               SPECIFIC MATERIAL S/N DATA SET
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  STR()      2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C               S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  ZZ()       1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
C               Z = F(STR,NF,NBND,MM)


C  INITIALIZE VARIABLES

      NP = 0

      DO 50 I = 1, MAXDAT
         ZZ(I) = 0.0
   50 CONTINUE


C  BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG
         MML = MM(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =',
     &       MML, ' NPTS =', NPTS(L)

         DO 200 K = 1, NPTS(L)
            NP = NP + 1
            ZZ(NP) = ALOG(STR(K,L)) + ALOG(NF(K,L)) * (1.0 / MML)
            IF (IOUT .EQ. 10) WRITE(8,*)'K =', K, ' NP =', NP, ' NF =',
     &         NF(K,L), ' STR =', STR(K,L), ' ZZ =', ZZ(NP)

            DO 300 LL = 2, L
```

```
              ZZ(NP) = ZZ(NP) + ALOG(NBND(LL-1))
     &                * ((1.0 / MM(LL-1)) - (1.0 / MM(LL)))
              IF (IOUT .EQ. 10) WRITE(8,*)'LL =', LL, ' NBND(LL-1) =',
     &           NBND(LL-1), ' MM(LL-1) =', MM(LL-1), ' MM(LL) =',
     &           MM(LL), ' ZZ =', ZZ(NP)
  300       CONTINUE

  200     CONTINUE

  100 CONTINUE

      RETURN
      END

C***********************************************************************


C  SUBROUTINE SMNVAR CALCULATES THE Sample MeaN and VARiance OF
C  Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  24AUG87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V3.3, V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE SMNVAR (NP, ZZ, MEANZ, SZ2)

C  INPUTS:   NP, ZZ
C  OUTPUTS:  MEANZ, SZ2

C     IMPLICIT NONE

      INTEGER MAXDAT

      PARAMETER (MAXDAT = 50)

      COMMON IOUT

      INTEGER I, IOUT, NP

      REAL    MEANZ, SZ2, ZZ(MAXDAT)

C                      LIST OF VARIABLES
C
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C               DATA SET
C  SZ2        SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  ZZ()       1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C               Z = F(STR,NF,NBND,MM)


C  INITIALIZE VARIABLES

      MEANZ = 0.0
      SZ2 = 0.0

C  CALCULATE THE MEAN OF ZZ(), MEANZ

      DO 100 I = 1, NP
         MEANZ = MEANZ + ZZ(I)
         IF (IOUT .EQ. 10) WRITE(8,*)'NP =', NP, ' I =', I,
     &      ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
  100 CONTINUE
      MEANZ = MEANZ / FLOAT(NP)
      IF (IOUT .EQ. 10) WRITE(8,*)' MEANZ =', MEANZ
```

```
C  CALCULATE THE VARIANCE OF ZZ(), SZ2

      DO 200 I = 1, NP
         SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
         IF (IOUT .EQ. 10) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
  200 CONTINUE
      SZ2 = SZ2 / FLOAT(NP - 1)
      IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 =', SZ2

      RETURN
      END


C*************************************************************************


C  SUBROUTINE KBETA CALCULATES k AND BETAo FROM THE SAMPLE MEAN AND
C  VARIANCE OF Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87       COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)

C  INPUTS:   MEANZ, SZ2
C  OUTPUTS:  K, BZERO

C     IMPLICIT NONE

      REAL     PI

      PARAMETER (PI = 3.1415926536)

      COMMON   IOUT

      INTEGER IOUT

      REAL     BZERO, K, MEANZ, SZ, SZ2


C                  LIST OF VARIABLES
C
C  BZERO        VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE
C                 SPECIFIC MATERIAL S/N DATA SET
C  IOUT         OUTPUT DUMP CONTROLLER
C  K            VALUE OF k - PARAMETER CHARACTERIZING SPECIFIC MATERIAL
C                 DATA BASE
C  MEANZ        SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  PI           SELF EXPLANATORY CONSTANT
C  SZ           SZ2 ** 0.5
C  SZ2          SAMPLE VARIANCE OF THE TRANSFORMED DATA,
C                 Z = F(STR, NF, NBND, MM)


C  PERFORM CALCULATIONS

      SZ = SZ2 ** 0.5

      BZERO = PI / (SZ * (6.0 ** 0.5))

      K = MEANZ

C  DATA DUMP STATEMENTS

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'SZ2 =', SZ2, ' SZ =', SZ
```

```
         WRITE(8,*) 'MEANZ =', MEANZ, ' K = ', K, ' BZERO =', BZERO
      ENDIF

      RETURN
      END


C***************************************************************************


C  SUBROUTINE FINDK CALCULATES THE VALUE OF K, WHERE A = K ** M FOR
C  EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  7JUN88
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C  INPUTS:   BZERO, K, MM, NBND, NUMREG
C  OUTPUTS:  BIGK

C     IMPLICIT NONE

      INTEGER MAXREG

      REAL    GAMMA

      PARAMETER (GAMMA = 0.57721566490, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    BIGK(0:MAXREG), BZERO, K, MM(0:MAXREG), NBND(0:MAXREG)


C                       LIST OF VARIABLES
C
C  BIGK()   1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C             FOR EACH REGION
C  BZERO    VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC
C             MATERIAL DATA BASE
C  GAMMA    EULER'S CONSTANT
C  IOUT     OUTPUT DUMP CONTROLLER
C  K        VALUE OF k - PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C             DATA BASE
C  L        CONTROLS DO LOOP FOR EACH REGION
C  MAXREG   MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()     1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()   1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C             REGIONS OF INTEREST
C  NUMREG   NUMBER OF REGIONS OF INTEREST


C  INITIALIZE VARIABLES

      DO 50 L = 0, MAXREG
         BIGK(L) = 0.0
   50 CONTINUE

C  CALCULATE K FOR REGION ONE

      BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
C      WRITE(7,*) 'REGION:  1,  K =', BIGK(1)
      IF (IOUT .EQ. 10) WRITE(8,*)'BZERO =', BZERO, ' k =', K,
     &   ' GAMMA =', GAMMA, ' BIGK(1) =', BIGK(1)

C  CALCULATE K FOR REMAINING REGIONS

      DO 100 L = 2, NUMREG
         BIGK(L) = BIGK(L-1) * NBND(L-1)
     &             ** ((1.0 / MM(L)) - (1.0 / MM(L-1)))
```

```
C          WRITE(7,*) 'REGION ', L, ' K =', BIGK(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' NBND(L-1) =',
     &       NBND(L-1), ' MM(L) =', MM(L), ' MM(L-1) =', MM(L-1),
     &         ' BIGK(L) =', BIGK(L)
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE FINDSB CALCULATES THE REGION 'TIE-POINTS' - THE STRESS
C  VALUES WHICH CORRESPOND TO THE "LIFE BOUNDARIES" ACCORDING TO THE
C  RANDOMLY SELECTED Ms, AND THE Ks CALCULATED FROM THE BETA AND k
C  CHARACTERIZING SPECIFIC MATERIAL
C  PROGRAMMER:  L. NEWLIN
C        DATE:  22DEC88
C     VERSION:  MATCHR V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C  INPUTS:    NUMREG, ZROREG, NBND, BIGK, MM
C  OUTPUTS:   SBND

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG, ZROREG

      REAL    BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
     &        SBND(0:MAXREG)


C             LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  IOUT      OUTPUT DUMP CONTROLLER
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  SBND()    1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C              REGION CONTAINED IN NBND()
C  ZROREG    ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO REGION


C  INITIALIZE SBND()

      DO 50 L = 0, MAXREG
         SBND(L) = 0.0
   50 CONTINUE

C   CALCULATE SBND(0) IF ZROREG = 0

      IF (ZROREG .EQ. 0) THEN
         SBND(0) = BIGK(1) * NBND(0) ** (-1.0 / MM(1))
      ENDIF

C  CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES
```

```
      DO 100 L = 1, NUMREG
         IF (NBND(L) .GE. 1.0E+36) THEN
            SBND(L) = 0.0
         ELSE
            SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
         ENDIF
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE KOMO CALCULATES Ko AND Mo FOR THE ZERO REGION (NO DATA
C  REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT
C  AT SZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
C  PROGRAMMER :  L. NEWLIN
C        DATE:  1AUG91
C     VERSION:  MATCHR V8.5   MATGRM V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
     &                 FACTR, NUMREG)

C  INPUTS:   SZERO, BIGK, MM, NBND, TRSBND, FACTR
C  OUTPUTS:  TRBIGK, MM, TRSBND

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG

      REAL     BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1         SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)


C                       LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C              EACH REGION
C  FACTR     SCALE FACTOR = PHI * KRATIO * Z
C  IOUT      OUTPUT DUMP CONTROLLER
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS
C  SCLK      ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
C  SZERO     STRESS TENSILE TEST POINT, So
C  TRBIGK() 1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP
C              SBND(0) < So FOR EACH TRIAL
C  TRSBND() 1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE
C              LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()
C              ADJUSTED BY VARIATION PARAMETERS FOR EACH TRIAL


      BIGK(0) = SZERO

      IF (TRSBND(0) .GT. SZERO) THEN
         SCLK = SZERO/TRSBND(0)
         DO 100 L = 0, NUMREG
```

```
            TRBIGK(L) = BIGK(L) * SCLK
            TRSBND(L) = TRSBND(L) * SCLK
 100     CONTINUE
      ELSE
         TRBIGK(0) = SZERO/FACTR
         MM(0) = MM(1) * ((ALOG (BIGK(1)) - ALOG (TRSBND(0))
     &           + ALOG (FACTR)) / (ALOG (SZERO) - ALOG (TRSBND(0))))
      ENDIF
C
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'SZERO = ', SZERO, ' BIGKo = ', TRBIGK(0)
         WRITE(8,*) 'FACTOR = ', FACTR, ' BIGK1 = ', TRBIGK(1)
         WRITE(8,*) 'MM1 = ', MM(1), ' MMo = ', MM(0)
      ENDIF

      RETURN
      END

C***********************************************************************


C   SUBROUTINE WORSTN FINDS THE WORST OF N FOR BOTH THE WEIBULL AND
C   LOGNORMAL DISTRIBUTIONS
C   PROGRAMMER:  L. NEWLIN
C         DATE:  14NOV90
C      VERSION:  MATCHR VB1.2, VB1.3
C                MATGRM VB1, VB1.1
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.

      SUBROUTINE WORSTN (RAND, NSYM, BZERO, MM, EPSW, EPSL)

C    INPUT:  RAND, NSYM, BZERO, MM
C   OUTPUT:  EPSW, EPSL
C ROUTINE:   RANDOM

C     IMPLICIT NONE

      COMMON    IOUT

      INTEGER  IOUT, MAXREG, NSYM

      PARAMETER (MAXREG = 3)

      REAL      BZERO, C0, C1, C2, D1, D2, D3, EPSW, EPSL, F,
     &          MM(0:MAXREG), P, P0, SIGMA, T, T2, T3, X

      DOUBLE PRECISION  RAND


C                 LIST OF VARIBLES
C
C   BZERO    WEIBULL SHAPE PARAMETER, BETA
C   C0, C1, C2, D1, D2, D3
C            COEFFICIENTS OF FUNCTION FOR LOGNORMAL DISTRIBUTION CALCULATIONS
C   EPSL     LOGNORMAL(0,SIGMA**2) WORST OF NSYM RANDOM VARIATE
C   EPSW     WIEBULL(BZERO) WORST OF NSYM RANDOM VARIATE
C   F        UNIFORM(0,1) RANDOM VARIATE, VALUE OF CDF
C   IOUT     OUTPUT DUMP CONTROLLER
C   MAXREG   MAXIMUM NUMBER OF REGIONS ALLOWED
C   MM()     1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C   NSYM     SYMMETRY NUMBER
C   P0       VALUE OF P USED TO CHECK/INSURE P>.5
C   P        INTERMEDIATE CALCULATION VARIABLE FOR LOGNORMAL DISTRIBUTION
C              CALCULATIONS
C   SIGMA    STANDARD DEVIATION OF THE LOGNORMAL DISTRIBUTION
C   T        INTERMEDIATE CALCULATION VARIABLE FOR LOGNORMAL DISTRIBUTION
C              CALCULATIONS
C   T2       EQUAL TO T**2
```

```
C  T3       EQUAL TO T**3
C  X        NORMAL(0,SIGMA**2) WORST OF NSYM RANDOM VARIATE

      C0 = 2.515517
      C1 = 0.802853
      C2 = 0.010328
      D1 = 1.432788
      D2 = 0.189269
      D3 = 0.001308

      SIGMA = 1.282550 * MM(1) / BZERO

      CALL RANDOM (F, RAND)

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'BZERO = ', BZERO, ' SIGMA = ', SIGMA
         WRITE(8,*) 'F = ', F, ' MM = ', MM(1)
      ENDIF

      EPSW = EXP ((LOG (- LOG(1.0 - F) / NSYM)
     &            - LOG (LOG (2.0))) * MM(1) / BZERO)

      IF (IOUT .EQ. 10) WRITE(8,*) 'EPSW = ', EPSW

      P0 = (1.0 - F) ** (1.0 / FLOAT (NSYM))

      IF (P0 .LE. 0.5) THEN
         P = P0
         T = SQRT (LOG (1.0 / P ** 2))
         IF (IOUT .EQ. 10) WRITE(8,*) 'P = ', P, ' T = ', T
         T2 = T * T
         T3 = T * T2
         X = T - ((C0 + C1 * T + C2 * T2)
     &           / (1.0 + D1 * T + D2 * T2 + D3 * T3))
         IF (IOUT .EQ. 10) WRITE(8,*) 'X = ', X
      ELSE
         P = 1.0 - P0
         T = SQRT (LOG (1.0 / P ** 2))
         IF (IOUT .EQ. 10) WRITE(8,*) 'P = ', P, ' T = ', T
         T2 = T * T
         T3 = T * T2
         X = - (T - ((C0 + C1 * T + C2 * T2)
     &           / (1.0 + D1 * T + D2 * T2 + D3 * T3)))
         IF (IOUT .EQ. 10) WRITE(8,*) 'X = ', X
      ENDIF
      EPSL = EXP (SIGMA * X)

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'BZERO = ', BZERO, ' SIGMA = ', SIGMA
         WRITE(8,*) 'F = ', F, ' EPSW = ', EPSW
         WRITE(8,*) 'P = ', P, ' T = ', T
         WRITE(8,*) 'T2 = ', T2, ' T3 = ', T3
         WRITE(8,*) 'X = ', X, ' EPSL = ', EPSL
      ENDIF

      RETURN
      END

C***********************************************************************

C  FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
C  BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB89
C     VERSION:  MATCHR V8.3, V8.4, V8.5 — FOR USE WITH PFM'S
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
```

```
      REAL FUNCTION GTLIFE (S, MM, LNA, LPHIM, KRATIO, LNZ, SBND,
     &                      ZROREG, NUMREG, SZERO)

C  INPUTS:   S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO
C  OUTPUTS:  GTLIFE

C     IMPLICIT NONE

      INTEGER IOUT, L, MAXREG, NUMREG, ZROREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      REAL     GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
     &         MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

C                       LIST OF VARIABLES
C
C  GETLIF    VALUE TO BE ASSIGNED TO GTLIFE - CYCLES TO FAILURE FOR
C              THE REQUIRED STRESS LEVEL
C  IOUT      OUTPUT DUMP CONTROLLER
C  KRATIO    RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L         CONTROLS DO LOOP FOR EACH REGION
C  LNA()     1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C  LNZ       NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  LPHIM()   1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
C              PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  S         VALUE OF STRESS (PSI) FOR WHICH A VALUE OF LIFE (CYCLES TO
C              FAILURE) IS REQUIRED
C  SBND()    1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
C              CONTAINED IN NBND()
C  SZERO     STRESS TENSILE POINT, So
C  TEMP      TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER
C              FLOWS
C  ZROREG    ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO REGION

      GETLIF = 0.0

C  CALCULATE CYCLES TO FAILURE

      IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
         GETLIF = 1.0
      ELSE
         DO 100 L = ZROREG, NUMREG
            IF (S .GT. SBND(L)) THEN
               TEMP = LNA(L) + LPHIM(L) + MM(L) * ( - ALOG(S)
     &                     +  ALOG (KRATIO) + LNZ)
               IF (TEMP .GT. 86.0) THEN
                  TEMP = 86.0
               ENDIF
               GETLIF = EXP (TEMP)
               GOTO 150
            ENDIF
  100    CONTINUE
      ENDIF
  150 CONTINUE

      GTLIFE = GETLIF

      RETURN
      END


C***********************************************************************
```

```
C  SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C  M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB88
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE SORTM (ALLM, NUMREG, NUM)

C   INPUTS:  ALLM, NUMREG, NUM
C  OUTPUTS:  ALLM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER I, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG

      PARAMETER (MAXMM = 20001, MAXREG = 3)

      LOGICAL INORDR

      REAL    ALLM(MAXMM, MAXREG), TEMP

C                    LIST OF VARIABLES
C
C  ALLM()     2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
C  I          CONTROLS INSERTION POINTER
C  INC        SORT INCREMENT VARIABLE
C  INORDR     FLAG TO INDICATE WHETHER SORT IS FINISHED
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXMM      MAXIMUM NUMBER OF M'S TO BE SORTED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUM        NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  TEMP       TEMPORARY SORTING VARIABLE


      DO 400 L = 1, NUMREG

    5    INC = NUM
   10    IF (INC .GT. 1) THEN
            INC = INC / 2
   20       INORDR = .TRUE.

            DO 300 I = 1, (NUM - INC)
               IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN
                  TEMP = ALLM(I,L)
                  ALLM(I,L) = ALLM(I + INC, L)
                  ALLM(I + INC, L) = TEMP
                  INORDR = .FALSE.
               ENDIF
  300       CONTINUE

            IF (.NOT. INORDR) GOTO 20
            GOTO 10
         ENDIF

  400 CONTINUE

      RETURN
      END
```

```
C***********************************************************************

C*******************************************************************
C
C  FUNCTION RAINF3 CALCULATES THE TIME (in missions) TO FAILURE FOR
C  THE GIVEN STRAIN-TIME HISTORY
C
C  PROGRAMMER:  L. SHIRAISHI, L. NEWLIN
C        DATE:  27MAR90
C     VERSION:  1.1 (BLDLCF V3.1, V3.2, V3.3, V3.4  MATCHR V8.4, V8.5)
C
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C*******************************************************************

      FUNCTION RAINF3 (SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM,
     &                 KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO)

C  INPUTS:    SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO,
C             LNZ, SBND, ZROREG, NUMREG, SZERO
C  OUTPUTS:   RAINF3

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER  MAXREG, MAXM

      PARAMETER (MAXREG = 3, MAXM = 50)

      INTEGER  I, INDEX(MAXM), IOUT, J, JMAX, K, M, N, NEWTOT, NUMREG,
     &         ZROREG

      REAL     CHKFT, E(MAXM), GTLIFE, INVLIF(MAXM), KRATIO,
     &         LIFE(MAXM), LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
     &         MM(0:MAXREG), PERIOD, RAINF3, S(MAXM), SBND(0:MAXREG),
     &         SEFF(MAXM), SEFFM(2, MAXM), SEFMAX, SP(MAXM),
     &         SRANGE(MAXM), SUMDAM, SZERO, TEST1(MAXM), TEST2(MAXM),
     &         TRUNC, WEXP


C                       LIST OF VARIABLES
C
C  RAINF3       TIME TO FAILURE FOR THE GIVEN TIME HISTORY
C
C input variables:
C
C  SEFF(M)      EFFECTIVE STRAINS BEFORE FILTERING/RAINFLOW
C  M            TOTAL NUMBER OF STRAIN DATA POINTS PER PERIOD
C  TRUNC        VALUE USED TO FILTER OUT NOISE
C  PERIOD       TIME IN SECONDS FOR ONE PERIOD
C  WEXP         WALKER EXPONENT
C
C
C intermediate variables:
C
C  MAXM         MAXIMUM NUMBER OF POINTS ALLOWED IN STRAIN-TIME HISTORY
C                 ARRAYS
C  SEFMAX       LARGEST EFFECTIVE STRAIN
C  JMAX         INDEX (LOCATION) OF SEFMAX IN SEFF()
C  I,J,K        COUNTERS FOR VARIOUS DO LOOPS
C  SP(M+1)      RESEQUENCED EFFECTIVE STRAINS; # OF PTS = M+1
C  INDEX(MAXM), TEST1(MAXM), TEST2(MAXM)
C                 INTERMEDIATE CALCULATION ARRAYS USED DURING FILTERING
C  S(NEWTOT)    FILTERED EFFECTIVE STRAINS
C  NEWTOT       TOTAL NUMBER OF EFFECTIVE STRAIN VALUES AFTER FILTERING
C  E()          HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
C  N            NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
C  SEFFM(2,N)   EFFECTIVE STRAINS AFTER RESEQUENCING/FILTERING/RAINFLOW
C                     SEFFM(1,I) = sigma max,eff,i
C                     SEFFM(2,I) = sigma min,eff,i
C  SRANGE(N)    SRANGE(I) = EQUIVALENT STRAIN RANGE FOR CYCLE I
```

```
C  GTLIFE       REAL FUNCTION THAT CALCULATES FATIGUE LIFE FOR A GIVEN STRAIN
C  LIFE(N)      LIFE(I) = CALCULATED LIFE FOR STRAIN LEVEL SRANGE(I)
C  INVLIF(N)    INVLIF(I) = 1/LIFE(I); DAMAGE FRACTION
C  SUMDAM       SUM OF ALL THE DAMAGE FRACTIONS
C  CHKFT        DUMMY VARIABLE USED TO PRINT OUT RAINF3 RESULT
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  LNA()        1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C  LNZ          NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  LPHIM()      1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
C                 PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()         1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  SBND()       1-D ARRAY CONTAINING THE STRAIN VALUES (%, R = - 1.0)
C                 CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C                 REGION CONTAINED IN NBND() CORRECTED BY PHI, KRATIO,
C                 AND LNZ
C  SZERO        STRAIN TENSILE POINT, So (%)
C  ZROREG       Zero REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                 REGION

C  dump input data
      if (iout.eq.20) then
        write(8,*)' rainf3 inputs'
        write(8,*)'m     :',m,'      period:',period

        write(8,*)'wexp :', wexp
        write(8,*)'numreg   :',numreg,'zroreg :',zroreg
        write(8,*)'szero   :',szero,'kratio   :',kratio,'lnz   :',lnz
        write(8,*)'lna(i),     mm(i),     lphim(i),     sbnd(i)'
        write(8,*) (lna(i), mm(i), lphim(i), sbnd(i), i=zroreg,numreg)
        write(8,*)' '
      endif

C  INITIALIZE ARRAYS

      DO 50 I = 1, MAXM
         SP(I) = 0.0
         S(I) = 0.0
         E(I) = 0.0
         SEFFM(1,I) = 0.0
         SEFFM(2,I) = 0.0
         SRANGE(I) = 0.0
         LIFE(I) = 0.0
         INVLIF(I) = 0.0
         INDEX(I) = 0
         TEST1(I) = 0.0
         TEST2(I) = 0.0
   50 CONTINUE


C********************* B E G I N    R E S E Q U E N C E ***********************
C  RESEQUENCE effective strains (needed for rainflow analysis);
C   largest effective strain is placed at beginning and end of SP(M+1)

C   find SEFMAX, the largest sigma,eff, and JMAX, its location within SEFF(M)
      SEFMAX = -1.0E+20
      DO 200 I=1,M
         IF ( SEFF(I) .GT. SEFMAX ) THEN
            SEFMAX = SEFF(I)
            JMAX = I
         ENDIF
  200 CONTINUE

C   assign all points from JMAX out, to the beginning of SP()
      DO 210 I = 1, M-JMAX+1
         J = JMAX-1 + I
         SP(I) = SEFF(J)
  210 CONTINUE

C   assign points before JMAX to the end of SP()
      J = 0
```

```
      DO 220 I = M-JMAX+2, M
         J = J + 1
         SP(I) = SEFF(J)
 220  CONTINUE
      SP(M+1) = SEFF(JMAX)
      if (iout.eq.20) then
         write(8,*)'sefmax:',sefmax,'    jmax:',jmax
         write(8,*)'sp(m+1):',(sp(i),i=1,m+1)
      endif

C********************** E N D   R E S Q U E N C E *****************************
C********************** B E G I N   F I L T E R *******************************
C  FILTER the resequenced effective strains, leaving only peaks and valleys
C   (excursions larger than TRUNC are deleted during rainflow counting) in
C   S(NEWTOT), where NEWTOT is the new number of points
C

      DO 300 I = 2, M
         TEST1(I) = SP(I-1) - SP(I)
         TEST2(I) = TEST1(I) * (SP(I) - SP(I+1))
  300 CONTINUE

C      if (iout .eq. 20) then
C         do 305 i = 2, m
C            write(8,*) 'test1 = ', test1(i), ' test2 = ', test2(i)
C  305    continue
C      endif

      K = 1
      INDEX(1) = 1

      DO 310 I = 2, M
         IF ((TEST1(I) .NE. 0) .AND. (TEST2(I) .LT. 0)) THEN
            K = K + 1
            INDEX(K) = I
         ENDIF
  310 CONTINUE

      NEWTOT = K + 1
      INDEX(NEWTOT) = M + 1

      DO 320 I = 1, NEWTOT
         K = INDEX(I)
         S(I) = SP(K)
  320 CONTINUE

      if (iout.eq.20) then
         write(8,*)'newtot:',newtot
         write(8,*)'s(newtot):',(s(i),i=1,newtot)
      endif

C************************** E N D   F I L T E R ******************************
C********************** B E G I N   R A I N F L O W **************************
C RAINFLOW ANALYSIS to identify cycles within effective strain data, S(NEWTOT);
C   places each cycle's max and min values into SEFFM(2,N)
C
C   counters: I counts # of cycles found, J counts how many S()'s counted,
C             K accumulates unmatched points
      I = 0
      J = 0
      K = 0
 400  CONTINUE
      J = J+1
      K = K+1
C   check J to avoid reading beyond end of filtered strain data
      IF ( J .GT. NEWTOT ) GOTO 499

C   read strain point into a holding array to be checked for cycles
      E(K) = S(J)
 410  IF ( K .LT. 3 ) GOTO 400
      IF ( ABS( E(K) - E(K-1) ) .LT. ABS( E(K-1) - E(K-2) ) ) GOTO 400
C    if not, then a cycle has been found, but we need to check for truncation
      IF (ABS (E(K-1) - E(K-2)) .GT. TRUNC) THEN
```

```
C        cycle is large enough to save
         I = I+1
         SEFFM(1,I) = AMAX1( E(K-1), E(K-2) )
         SEFFM(2,I) = AMIN1( E(K-1), E(K-2) )
      ENDIF
C   discard points K-1 and K-2, and decrement the counter of unmatched points
      E(K-2) = E(K)
      K = K-2
C   return for more counting
      GOTO 410

  499 CONTINUE
C   N equals the final number of cycles found
      N = I

      if (iout.eq.20) then
         write(8,*)'N :',n
         write(8,*)'seffm(2,n):'
         do 12 i=1,n
            write(8,*) seffm(1,i), seffm(2,i)
 12      continue
      endif

      IF (N .EQ. 0) THEN
C        truncation filter value too large — no cycles left
         SUMDAM = 1.0E-36
         GOTO 710
      ENDIF

C
C********************** E N D    R A I N F L O W **************************

C  calculate equivalent strain range
C
      DO 500 I=1,N
         SRANGE(I) = (SEFFM(1,I) - SEFFM(2,I))
     &               * ((SEFFM(1,I) - SEFFM(2,I)) / (2.0 * SEFFM(1,I)))
     &               ** (WEXP - 1.0)
 500  CONTINUE
      if (iout.eq.20) write(8,*)'srange(n) :',(srange(i),i=1,n)
      if (iout.eq.25) write(8,*) (srange(i),i=1,n), ' ',
     &   exp(lphim(1)/mm(1))

C  calculate lives and damage fractions: LIFE(N) and INVLIF(N)
C
      DO 600 I=1,N
         LIFE(I) = GTLIFE (SRANGE(I), MM, LNA, LPHIM, KRATIO, LNZ,
     &                     SBND, ZROREG, NUMREG, SZERO)
 600  CONTINUE
      DO 650 I=1,N
         INVLIF(I) = 1.0 / LIFE(I)
 650  CONTINUE
      if (iout.eq.20) then
         do 14 i=1,n
            write(8,*)'life(n):',life(i),'      invlif(n):',invlif(i)
 14      continue
      endif

C  Miner's Rule — sum the damage fractions
C
      SUMDAM = 0.0
      DO 700 I=1,N
         SUMDAM = SUMDAM + INVLIF(I)
  700 CONTINUE
  710 CONTINUE
      if (iout.eq.20) write(8,*)'sumdam:',sumdam

C  calculate fatigue life (time to failure)
C
      RAINF3 = PERIOD / SUMDAM

      if (iout.eq.15) then
         chkft=period/sumdam
```

```
   write(8,*)' rainf3 life',chkft
   write(8,*)
endif

RETURN
END
```

Section 7.3
High Cycle Fatigue Failure Program BLDHCF

The program tree structures, list of subprograms, descriptions of the key variables, and the FORTRAN source listing for the high cycle fatigue analysis code BLDHCF are given here. The pertinent HCF methodology is given in Section 4. The overall description of the program and the flowcharts are given in Section 5.3.

7.3.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for BLDHCF, using Uniform variation on the materials shape parameter *m*, is given in Figure 7.3-1, while the tree structure for the truncated Normal case is given in Figure 7.3-2. In both trees, those subprograms not "shadow-boxed" are part of the materials characterization model. The program, subprogram, and file names are indicated by UPPERCASE letters.

7.3.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7.3-1. The section numbers where the subprograms are described by means of flowcharts are given next to the names.

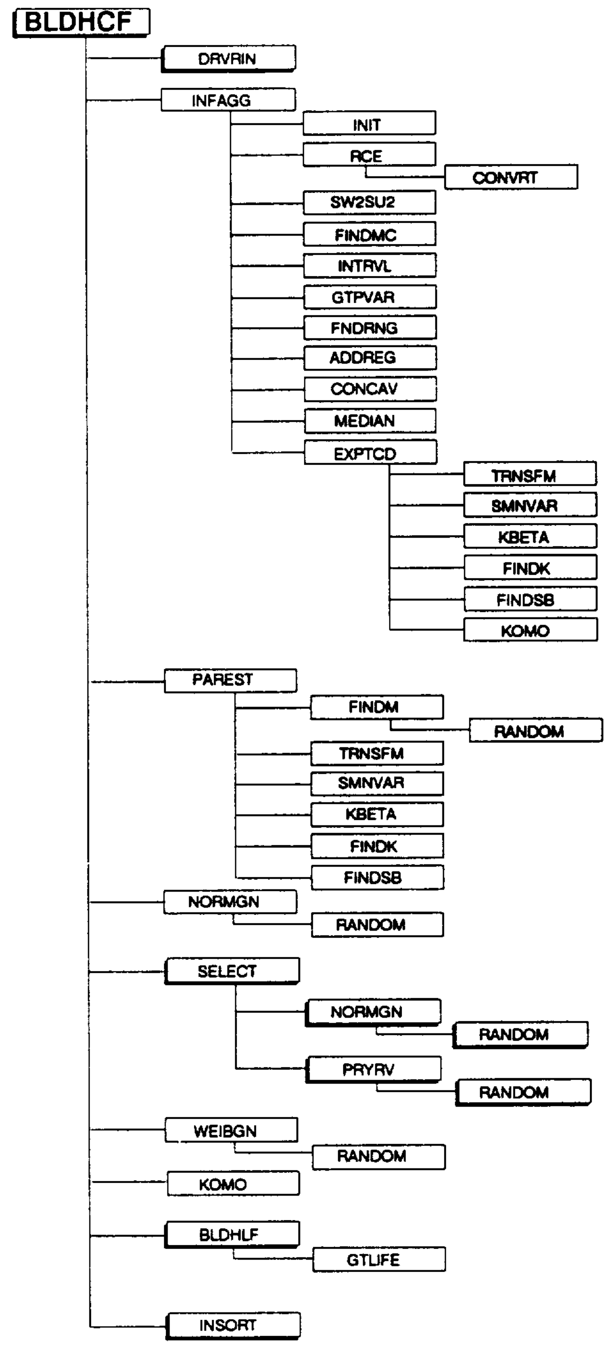


Figure 7.3-1 Tree Structure for Program BLDHCF for the Uniform Variation in Materials Shape Parameter *m*

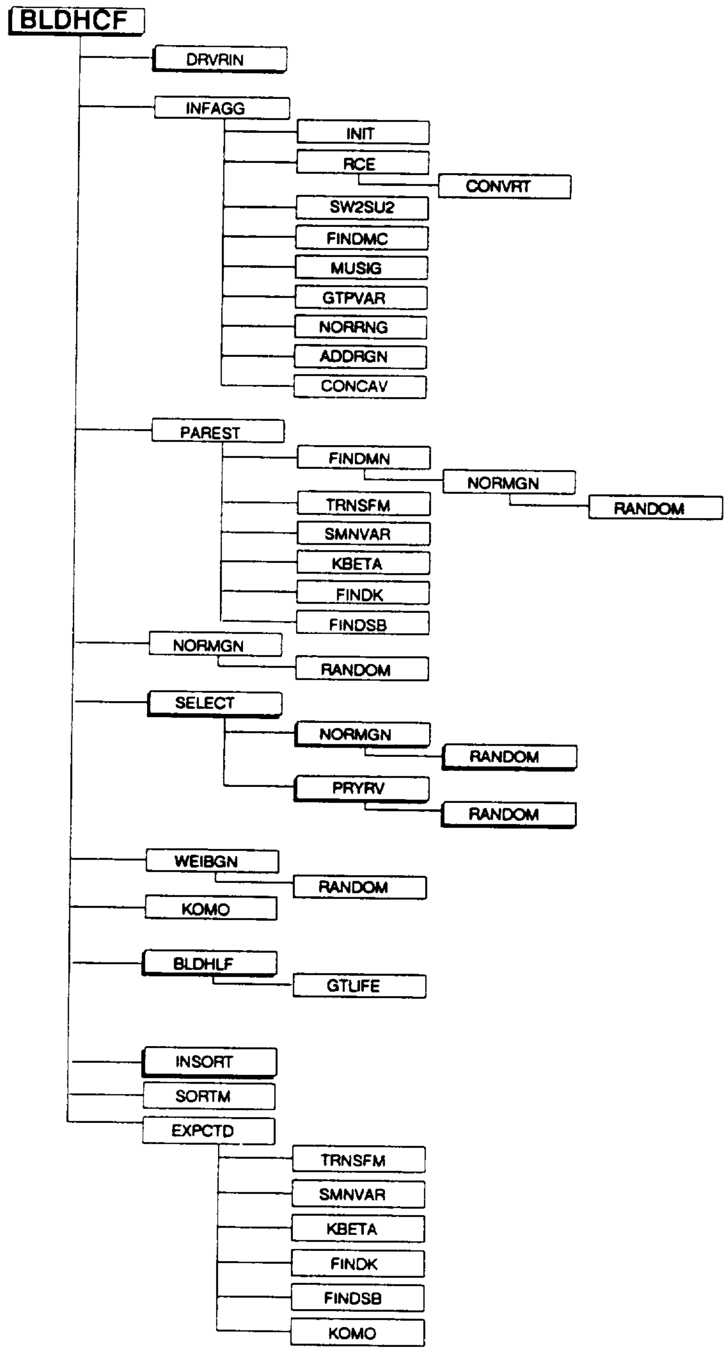


Figure 7.3-2 Tree Structure for Program BLDHCF for the Truncated Normal Variation in Materials Shape Parameter *m*

Table 7.3-1 List of Subprograms For Program BLDHCF
(Footnotes are at the end of the table)

NAME	SECTION	PURPOSE
ADDREG[1]	4.1.3.9*	Adds the *m* ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.
ADDRGN[1]	4.1.3.15*	Adds the *m* ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.
BLDHCF	5.3.2.1	The main routine that controls the logical flow of the high cycle fatigue turbine blade program.
BLDHLF	5.3.2.4	Performs the calculations of the driver transformation and the fatigue life.
CONCAV[2]	4.1.3.10*	Adjusts the upper bound of the posterior ranges on *m* to be consistent with concavity constraints.
CONVRT[3]	4.1.3.3*	Transforms stress data to equivalent zero-mean stresses with stress ratio of −1.0.
DRVRIN	5.3.2.2	Reads the driver distributions and other structural and geometric parameters from BLDHCD and echoes the data to BLDHCO.
EXPCTD[4]	4.1.3.12*	Calculates the median S/N curve parameters from the results of the information aggregation calculations.
FINDK	4.1.5.6*	Calculates the value of the location parameter K (where $A = K^m$) for each life region by using Equations 2-37 and 2-41 of [1].
FINDM[5]	4.1.5.1*	Obtains the value of *m* for each life region by adjusting the range (to ensure concavity) and then sampling from the Uniform distribution over the appropriate *m* range.
FINDMC	4.1.3.5*	Calculates the *m* range implied by the constraint on the coefficient of variation of fatigue strength, *C*, for each life region, by using Equations 2-28 through 2-32 of [1].
FINDMN[5]	4.1.5.2*	Obtains the value of *m* for each life region by sampling from the appropriate truncated Normal distribution on *m*.
FINDSB	4.1.5.7*	Calculates the life region "tie-points" or stress values which correspond to the "life boundaries," conditional on the randomly selected *m* for each region. Also calculates *K*, characterizing the specific material S/N data set, which is a function of β_o and *k*.
FNDRNG[6]	4.1.3.8*	Combines the 95% confidence interval; J_o, with the implicit and explicit constraints on *m*, to obtain posterior credibility ranges on *m* for each life region.
GTLIFE	4.1.8*	Calculates the cycles to failure for a particular stress, based upon the materials characterization model S/N curve of Equation 2-48 of [1].

Table 7.3-1 List of Subprograms For Program BLDHCF (Cont'd)

NAME	SECTION	PURPOSE
GTPVAR	4.1.3.7*	Calculates σ^2, the extent of departures from the multiple heat median S/N curve warranted by the information available, by using Equation 2-49 of [1].
INFAGG[7]	4.1.3*	Controls the logical flow for the information aggregation portion of the materials characterization model.
INIT	4.1.3.1*	Initializes the entries of the arrays used in the information aggregation subroutine, INFAGG, to zero.
INSORT	5.B*	Performs an insertion sort for the lowest fifty percent of the lives calculated.
INTRVL	4.1.3.6*	Calculates the 95% confidence intervals I_o for C, and J_o for m, for each region by using Equations 2-24 through 2-26 of [1].
KBETA	4.1.5.5*	Calculates k and β_o from the sample mean and variance of Z, where Z is a function of stress, life, the life region boundaries, and the m's by using Equation 2-42 of [1].
KOMO[8]	4.1.6*	Calculates K_o and m_o for the zero region, the no data region to the left of the first data region. Extends the S/N curve consistent with the tensile point at S_o.
MEDIAN	4.1.3.11*	Calculates the median values of m, based on the posterior credibility ranges of m, by using Equation 2-34 of [1].
MUSIG[9]	4.1.3.13*	Calculates the posterior Normal distribution parameters, mean m_* and standard deviation σ_*, for each life region of the S/N curve.
NORMGN[10]	4.4.3*	Generates Normal(μ, σ^2) random variates.
NORRNG[6]	4.1.3.14*	Combines the implicit and explicit constraints on m to obtain the posterior credibility ranges of m for each life region.
PAREST[11]	4.1.5*	Controls the logical flow for the parameter estimation model portion of the materials characterization model.
PRYRV[12]	7.6.6*	Generates the Uniform(a, b) and Uniform(c, d) pair of independent random variates.
RANDOM[12]	4.4.2*	Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.
RCE	4.1.3.2*	Reads the data from BLDHCD and RELATD; calls CONVRT to transform the stress data to a stress ratio of −1.0; and echoes the data to BLDHCO and RELATO. RCE also breaks S/N data sets into regions as specified by the user.
SELECT	5.3.2.3	Performs the driver selection.
SMNVAR	4.1.5.4*	Calculates the sample mean and variance of Z, where Z is a function of stress, life, the life region boundaries, and the m's, by using Equation 2-42 of [1].

Table 7.3-1 List of Subprograms For Program BLDHCF (Cont'd)

NAME	SECTION	PURPOSE
SORTM[13]	4.1.10*	Sorts the *m* values in increasing order for each life region for the truncated Normal distribution case.
SW2SU2	4.1.3*	Calculates the residual variances from the *Y* on *X* and *X* on *Y* regressions for each life region where $Y = \ln(\textit{Endurance cycles})$ and $X = \ln(\textit{Stress})$ by using Equations 2-20 and 2-21 of [1]; to be used in the credibility range calculations.
TRMNAT	4.1.11*	Performs premature program termination when required.
TRNSFM[14]	4.1.5.3*	Performs the calculations necessary to transform the specific material S/N data into the variable *Z*, where *Z* is a function of stress, life, the life region boundaries, and the *m*'s.
WEIBGN	4.4.6*	Generates Weibull(β, $\eta(\beta)$) random variates.

---

* See [1].

1 No data regions to the right are discussed in [1], Page 2-17.

2 Concavity constraints are discussed in [1], Pages 2-13 through 2-14.

3 The stress transformation is discussed in [1], Page 2-7.

4 The median S/N curve parameter estimation calculations are described in [1], Pages 2-15 through 2-18.

5 Selection of the $\{m_j\}$ parameters is discussed in [1], Page 2-15.

6 Combining information to obtain the posterior credibility ranges on *m* is discussed in [1], Page 2-13.

7 The information aggregation calculations are discussed in [1], Pages 2-6 through 2-14.

8 Extension of the S/N curve to the left is discussed in [1], Page 2-17.

9 Calculation of the truncated Normal distribution parameters is discussed in [1], Page 2-14.

10 The Normal distribution is discussed in [1], Page 2-23.

11 The parameter estimation calculations are discussed in [1], Pages 2-15 through 2-18.

12 The Uniform distribution is discussed in [1], Page 2-23.

13 The need for saving *m*'s is discussed in [1], Page 2-15.

14 The S/N data transformation is discussed in [1], Page 2-16.

7.3.3 Description of Variables

A list of variables used in the ATD-HPOTP first and third stage turbine blade HCF code, BLDHCF, is given in Table 7.3-2. The variable names are indicated by **BOLD UPPERCASE** letters; the variable "type" can be interpreted as follows: INT is a standard integer variable; RE is a standard real variable; and DRE is a double precision variable. The various array dimensions are defined by using the following parameters: **MAXBLF**, **MAXDAT**, **MAXLIF**, **MAXMM**, and **MAXREG**.

Table 7.3-2 List of Variables For Program BLDHCF (Footnotes are at the end of the table)

VARIABLE NAME	TYPE	DESCRIPTION
A0, A1	RE	The coefficients for the flow rate $\dot{m}$ response surface function (performance balance characterization).
ALLM(MAXMM, MAXREG)	RE	2-D array containing the materials model shape parameters (m's) for each life region which are to be used in the truncated Normal median S/N curve calculation.[1]
B0, B1	RE	The coefficients for the enthalpy change Δh response surface function (performance balance characterization).
BIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter K, where $A = K^m$, given in Equation 2-12 of [1].
BIGK1	RE	Dummy variable used during calls to subroutine EXPCTD, equal to **BIGK(1)**.
BLDHLF	RE	Real function that performs the calculations of the driver transformation and fatigue life, and returns the fatigue life (sec).
BLFPER(MAXBLF)	RE	1-D array containing user specified B-lives which are obtained from the simulated failure distribution. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent: e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.
BLFPOS(MAXBLF)	INT	1-D array containing the indices for the array variable **LIFE( )** corresponding to the user-requested simulated failure distribution B-lives contained in variable **BLFPER( )**.

Table 7.3-2 List of Variables For Program BLDHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
BZERO	RE	Estimate of Weibull distribution shape parameter β_o, that characterizes the intrinsic variation of the S/N data set, by using Equation 2-11 of [1].
C	RE	*C* (in.) in Equation 4-1, the randomly selected distance from the turbine blade neutral axis.
C0, C10, C11, C20, C21	RE	The coefficients for the damper effectiveness response surface.
CM	RE	Mean, μ, of Normally distributed *C*, the distance from the turbine blade neutral axis (in.), given in Equation 4-1.
CS	RE	Standard deviation, σ, of Normally distributed *C*, the distance from the turbine blade neutral axis (in.), given in Equation 4-1.
DELTAH	RE	Δh (Btu/lbm) in Equation 4-1, the enthalpy change across the turbine stage.
DUM	RE	Dummy variable.
FACTR	RE	Equal to FACTOR = **PHI * KRATIO * Z**. Used by the materials model.
FIFTY	RE	Variable used to access the fifty-percent point in the **LIFE( )** array.
FTU	RE	Material ultimate strength (psi).
FTY	RE	Material yield strength (psi).
GTLIFE	RE	Function given by Equation 2-48 of [1] that calculates the fatigue cycles to failure at a given stress.
I	INT	Controls inner DO loop.
IMIN	RE	I_{min} (in.^{4}) in Equation 4-1, the minimum moment of inertia of the turbine blade cross section.
IOUT	INT	Output dump controller.
J	INT	Controls DO loop for each B-life.[2]
K	INT	Controls outer DO loop.
KRATIO	RE	Ratio of *MED K*/MED K* in Equation 2-48 of [1]. **KRATIO** is constant over life regions for the materials model.
L	INT	Controls DO loop for each life region of the S/N curve.

Table 7.3-2 List of Variables For Program BLDHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LAMB	RE	λ_B, the randomly selected turbopump performance balance model accuracy factor.
LAMBA	RE	Uniform distribution lower bound of λ_B.
LAMBB	RE	Uniform distribution upper bound of λ_B.
LAMD	RE	λ_D, the randomly selected damper coefficient of friction model accuracy factor.
LAMDA	RE	Uniform distribution lower bound of λ_D.
LAMDB	RE	Uniform distribution upper bound of λ_D.
LIFE	RE	L, the fatigue life in seconds.
LIFE(MAXLIF)	RE	1-D array containing values of the lives generated by program BLDHCF. The lives are sorted values for the left-hand tail simulated failure distribution.
LNA(0:MAXREG)	RE	1-D array containing values of ln(A) = ln(**BIGK**) * **MM** for each life region of the S/N curve.
LNZ	RE	ln(Z) in Equation 2-48 of [1], the Normal(0, **PVAR**) random variate for the materials process variation aspect of the materials model.
LPHIM(0:MAXREG)	RE	1-D array containing values of ln(**PHI**) * **MM** for each life region of the S/N curve.
M	INT	Controls symmetry DO loop.
MAXBLF	INT	Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is 10.[2]
MAXDAT	INT	Maximum number of points per data set per region allowed for the S/N curve. The maximum number of data points per set allowed is 50.
MAXLIF	INT	Maximum number of fatigue lives allowed for the simulated failure distribution. The maximum number of fatigue lives to be saved is 10,000.
MAXMM	INT	Maximum number of m's to be saved and sorted for the truncated Normal median S/N curve.[1] The maximum number of m's is 20,000.
MAXREG	INT	Maximum number of life regions allowed for the S/N curve. The maximum number of regions is 3.
MCOUNT	INT	Counts number of m's to be used to calculate the median S/N curve for the truncated Normal distribution case.[1]

Table 7.3-2 List of Variables For Program BLDHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
MD	RE	m_d, the damper mass (lbm).
MDOT	RE	$\dot{m}$ (lbm/sec) in Equation 4-1, the fluid mass flow rate.
MEDM(MAXMM)	RE	1-D array containing the empirical median *m* for each life region of the S/N curve.[3]
MID	INT	Pointer to the median *m* values in array **SORTM( )** for the truncated Normal median S/N curve. Value of half of **MCOUNT**.
MINPHI	RE	Value of min(**PHI**), the minimum of **NSYM** draws of the materials scatter parameter φ.
MM(0:MAXREG)	RE	m_j in Equation 2-12 of [1], the 1-D array containing randomly selected values of the materials model shape parameter *m* for each life region of the S/N curve.
MPROC	INT	Materials PROCess variation. Controls materials process variation. A value of 0 indicates no materials process variation, while a value of 1 indicates that materials process variation should be included.[4]
MRW2	RE	$m_d \, r_d \, \omega^2$, the damper normal load (lbf).
MU(MAXREG)	RE	1-D array containing the posterior Normal distribution mean[5] of the materials shape parameter *m* for each life region of the truncated Normal S/N curve.
MW	RE	m_w in Equation 4-7, the randomly selected characteristic exponent for the Walker relation.
MWA	RE	Uniform distribution lower bound of m_w.
MWB	RE	Uniform distribution upper bound of m_w.
NB	INT	N_b in Equation 4-1, the number of rotor blades.
NBLIFE	INT	Number of B-lives to be obtained from the simulated failure distribution.[2]
NBND(0:MAXREG)	RE	$N*_{i,\,i+1}$ in Equation 2-35 of [1], the 1-D array containing upper bounds for the **NUMREG** life regions of interest for the specific material S/N data set.
NEWLIF	RE	Fatigue life value (missions) returned from call to function BLDHLF.
NF	RE	N_f, the fatigue life in cycles.

Table 7.3-2 List of Variables For Program BLDHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
NF(MAXDAT, MAXREG)	RE	2-D array containing values from the array **RAWNF( )** for the specific material S/N data set partitioned into life regions.
NHYPER	INT	The outer loop size.
NLIFE	INT	The inner loop size.
NLIFET	INT	Total number of lives calculated by program BLDHCF. Value of **NHYPER * NLIFE**.
NMED	INT	Controls S/N curve median calculation for the truncated Normal distribution case. A value of 0 indicates that the user does not desire a median calculation or that the Uniform distribution case is being used; while a value of 1 indicates that the user desires the median calculation to be performed.
NPTS(MAXREG)	INT	1-D array containing the number of points per life region for the specific material S/N data set.
NS	INT	N_s, the number of stator blades.
NSYM	INT	Symmetry number, usually equal to the multiplicity of the modeling unit in the component.
NUMREG	INT	R in Equation 2-11 of [1], the number of life regions of interest in the S/N curve.
PHI	RE	φ in Equation 2-11 of [1], the material's intrinsic variation, or scatter, given by a Weibull(β_o, $\eta_o(\beta_o)$) random variate.
PSIG	RE	σ in Equation 2-48 of [1], the value of SQRT(**PVAR**).
PVAR	RE	σ^2 in Equation 2-48 of [1], characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.
R	RE	R in Equation 4-6, the stress ratio.
RAND	DRE	Random number seed.
RANGEM(2, MAXREG)	RE	2-D array containing values of the posterior credibility ranges on the materials model shape parameter m for each life region in the S/N curve. **RANGEM(1,L)** is the lower bound and **RANGEM(2,L)** is the upper bound in region L.[6]
RAVG	RE	r_{avg} (in.) in Equation 4-1, the randomly selected average turbine blade radius relative to the shaft center.

Table 7.3-2 List of Variables For Program BLDHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
RAVGM	RE	Mean, μ, of Normally distributed r_{avg}, the average turbine blade radius relative to the shaft center (in.).
RAVGS	RE	Standard deviation, σ, of Normally distributed r_{avg}, the average turbine blade radius relative to the shaft center (in.).
RD	RE	r_d, the randomly selected damper radius (in.).
RDM	RE	Mean, μ, of Normally distributed r_d, the damper radius (in.).
RDS	RE	Standard deviation, σ, of Normally distributed r_d, the damper radius (in.).
RPM	RE	ω (rpm) in Equation 4-1, the randomly selected steady state rotor speed.
RPMM	RE	Mean, μ, of Normally distributed ω, the steady state rotor speed (rpm).
RPMS	RE	Standard deviation, σ, of Normally distributed ω, the steady state rotor speed (rpm).
RROOT	RE	r_{root} (in.) in Equation 4-1, the randomly selected turbine blade root radius relative to the shaft center.
RROOTM	RE	Mean, μ, of Normally distributed r_{root}, the turbine blade root radius relative to the shaft center (in.).
RROOTS	RE	Standard deviation, σ, of Normally distributed r_{root}, the turbine blade root radius relative to the shaft center (in.).
SALT	RE	σ_{ALT} (psi) in Equation 4-3, the alternating stress.
SBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = −1.0, corresponding to the "life boundary" values for each life region of the S/N curve contained in array **NBND( )**.
SBRM	RE	$\overline{\sigma_{BR}}$ (psi) in Equation 4-1, the blade root mean stress.
SDSUD	RE	σ_D / σ_{UD} (psi) in Equation 4-3, the ratio of the damped blade vibratory stress to the undamped blade vibratory stress.
SEQ	RE	σ_{EQ} (psi) in Equation 4-7, the equivalent zero-mean stress amplitude.

Table 7.3-2 List of Variables For Program BLDHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
SIG(MAXREG)	RE	1-D array containing the posterior Normal distribution standard deviation[7] of the materials model shape parameter *m* for each life region of the truncated Normal S/N curve.
SMAX	RE	σ_{MAX} (psi) in Equation 4-4, the maximum or peak stress.
SMEAN	RE	σ_{MEAN} (psi) in Equation 4-2, the mean stress.
SMIN	RE	σ_{MIN} (psi) in Equation 4-5, the minimum or trough stress.
STR(MAXDAT, MAXREG)	RE	2-D array containing stress points with stress ratio = −1.0, for the specific material S/N data set partitioned into life regions.
SUD	RE	σ_{UD} (psi) in Equation 4-3, the undamped blade vibratory stress.
SZERO	RE	Stress tensile test point, S_o (psi).[8]
TRBIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter *K* consistent with the tensile point S_o.[8]
TRSBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = −1.0, corresponding to the "life boundary" values for each region of the S/N curve contained in array **NBND( )** for each **PHI** draw consistent with the tensile point S_o.[8]
VARY	INT	Controls type of S/N curve variation desired. A value of 0 indicates that no variation is required; a value of 1 means that intrinsic materials variation only is desired; a value of 2 indicates that the user desires a Uniform distribution on *m*; while a value of 3 indicates that a truncated Normal distribution is desired.
Z	RE	*Z* in Equation 2-48 of [1], the randomly selected process variation shift factor given by a Lognormal(0, **PVAR**) random variate.
ZROREG	INT	ZeRO REGion, the variable permits the inclusion of the tensile point S_o. The value of 0 implies a DO loop from zero to **NUMREG**, while a value of 1 causes the DO loop to be executed from one to **NUMREG**.[8]

1 The need for saving m's is discussed in [1], Page 2-15.

2 See variable **BLFPER( )** for a description of B-life.

3 The median S/N curve for the truncated Normal case is discussed in [1], Page 2-15.

4 See [1], Section 2.1.2.3, for a discussion on process variation in materials.

5 m_* of the posterior density of m is discussed in [1], Page 2-14.

6 The posterior credibility ranges $\pi(m)$ are discussed in [1], Page 2-13.

7 σ_* of the posterior density of m is discussed in [1], Page 2-14.

8 Extension of the S/N curve to the left using the tensile point is discussed in [1], Page 2-17.

7.3.4 Program BLDHCF Listing

BLDHCF Version 1.1

Program BLDHCF Listing Temporal Order, Uniform Distribution

Program BLDHCF Listing Temporal Order, Truncated Normal Distribution

```
C***********************************************************************
C  PROGRAM BLDHCF CONTROLS THE FLOW OF LOGIC OF THE HIGH CYCLE
C  FATIGUE ANALYSIS OF THE TURBINE BLADE FOIL PROBLEM
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20APR92
C     VERSION:  1.1 (MATCHR V8.5, INSORT V2.1)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      PROGRAM BLDHCF

C  SUBPROGRAMS:  DRVRIN, INFAGG, PAREST, NORMGN, SELECT, WEIBGN,
C                TRMNAT, BLDHLF, INSORT, SORTM, EXPTCD
C        FILES:  1:BLDHCD-OLD; 3:BLDHCO-NEW; 5:RELATD-OLD; 6:RELATO-NEW;
C                7:DUMP-NEW;   8:IOUTPR-NEW; 9:LOWLIF-NEW;
C                NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

C     IMPLICIT NONE

      INTEGER  MAXBLF, MAXDAT, MAXLIF, MAXMM, MAXREG

      PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLIF = 10000,
     &           MAXMM = 20001, MAXREG = 3)

      COMMON   IOUT

      INTEGER  BLFPOS(MAXBLF), I, IOUT, J, K, L, M, MCOUNT, MID,
     &         MPROC, NB, NBLIFE, NHYPER, NLIFE, NLIFET, NMED,
     &         NPTS(MAXREG), NS, NSYM, NUMREG, VARY, ZROREG

      DOUBLE PRECISION  RAND

      REAL     A0, A1, ALLM(MAXMM, MAXREG), B0, B1, BIGK(0:MAXREG),
     &         BIGK1, BLDHLF, BLFPER(MAXBLF), BZERO, C, C0, C10, C11,
     &         C20, C21, CM, CS, FACTR, FIFTY, FTU, FTY, IMIN, KRATIO,
     &         LAMB, LAMBA, LAMBB, LAMD, LAMDA, LAMDB, LIFE(MAXLIF),
     &         LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MD, MEDM(MAXREG),
     &         MINPHI, MM(0:MAXREG), MU(MAXREG), MW, MWA, MWB,
     &         NBND(0:MAXREG), NEWLIF, NF(MAXDAT, MAXREG), PHI, PSIG,
     &         PVAR, RANGEM(2, MAXREG), RAVG, RAVGM, RAVGS, RD, RDM,
     &         RDS, RPM, RPMM, RPMS, RROOT, RROOTM, RROOTS,
     &         SBND(0:MAXREG), SIG(MAXREG), STR(MAXDAT, MAXREG), SZERO,
     &         TRBIGK(0:MAXREG), TRSBND(0:MAXREG), Z

C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

      OPEN (1,  FILE = 'BLDHCD', STATUS = 'OLD')
      OPEN (3,  FILE = 'BLDHCO', STATUS = 'NEW')
      OPEN (7,  FILE = 'DUMP',   STATUS = 'NEW')
      OPEN (8,  FILE = 'IOUTPR', STATUS = 'NEW')
      OPEN (9,  FILE = 'LOWLIF', STATUS = 'NEW')

      READ(1,*) RAND
      WRITE(8,*) '                          RANDOM NUMBER SEED =', RAND
      READ(1,*) IOUT
      WRITE(8,*) '            IOUT (MATCHR = 10, BLDHCF = 15) =', IOUT
      READ(1,*) NLIFE
      WRITE(8,*) '                             INNER LOOP SIZE =', NLIFE
      READ(1,*) NHYPER
      WRITE(8,*) '                             OUTER LOOP SIZE =',NHYPER
      READ(1,*) NSYM
      WRITE(8,*) '                             SYMMETRY NUMBER =', NSYM
      READ(1,*) VARY
      WRITE(8,*) '              TYPE OF S/N VARIATION DESIRED '
      WRITE(8,*) ' (0-NONE; 1-INTRINSIC; 2-UNIFORM; 3-NORMAL) =',VARY
      READ(1,*) NMED
      WRITE(8,*) '      NORMAL MEDIAN CURVE (0 - NO, 1 - YES) =',NMED
      READ(1,*) MPROC
      WRITE(8,*) '         MATERIALS PROCESS VARIATION DESIRED'
      WRITE(8,*) '                            (0 - NO, 1 - YES) =', MPROC
```

```
      IF ((VARY .LT. 0) .OR. (VARY .GT. 3)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF S/N VARIATION DESIRED'
         CALL TRMNAT
      ENDIF
      IF ((NMED .NE. 0) .AND. (NMED .NE. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID RESPONSE TO NORMAL MEDIAN ',
     &              'CURVE QUESTION'
         CALL TRMNAT
      ENDIF

      IF ((MPROC .LT. 0) .OR. (MPROC .GT. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF MATERIALS PROCESS ',
     &              'VARIATION DESIRED'
         CALL TRMNAT
      ENDIF

      READ(1,*) NBLIFE
      IF (NBLIFE .GT. 0) READ(1,*) (BLFPER(J), J = 1, NBLIFE)
C ** CALL DRVRIN TO READ DATA FROM BLDHCD AND ECHO DATA TO BLDHCO

      CALL DRVRIN (RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS, CM, CS,
     &             RDM, RDS, LAMBA, LAMBB, LAMDA, LAMDB, MWA, MWB,
     &             IMIN, MD, NB, NS, A0, A1, B0, B1, C0, C10, C11,
     &             C20, C21)

C ** CALL INFAGG TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECT
C       OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS

      CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, ZROREG, NUMREG,
     &             NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR)

      IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)

      MCOUNT = 0

C ** INITIALIZE VARIABLES

      DO 35 K = 1, MAXLIF
         LIFE(K) = 1.0E+36
   35 CONTINUE

      NLIFET = NHYPER * NLIFE

C ** OUTER LOOP - THIS LOOP SAMPLES HYPER-PARAMETER SETS

      DO 150 K = 1, NHYPER

C ** CALL PAREST TO PERFORM THE PARAMETER ESTIMATION ASPECT OF THE
C       MATERIALS CHARACTERIZATION MODEL CALCULATIONS

         CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG,
     &                RAND, NBND, STR, BIGK, BZERO, MM, SBND)

C ** OBTAIN MATERIALS PROCESS VARIATION IF DESIRED

         CALL NORMGN (RAND, 0.0, PSIG, LNZ)

         IF (MPROC .EQ. 1) THEN
            Z = EXP (LNZ)
         ELSE
            KRATIO = 1.0
            Z = 1.0
            LNZ = 0.0
         ENDIF

         MCOUNT = MCOUNT + 1
         DO 175 L = 1, NUMREG
            ALLM(MCOUNT, L) = MM(L)
  175    CONTINUE

C ** INNER LOOP - THIS LOOP GENERATES BLADE FAILURE TIMES

         DO 200 I = 1, NLIFE
```

```
C ** INITILIZE S/N CURVE PARAMETERS

            DO 225 L = 0, MAXREG
               LNA(L) = 0.0
               LPHIM(L) = 0.0
               TRSBND(L) = 0.0
  225       CONTINUE

C ** CALL SELECT TO "SELECT" DRIVERS FOR CALCULATING LIFE

            CALL SELECT (RAND, RPM, RPMM, RPMS, RROOT, RROOTM,
     &                   RROOTS, RAVG, RAVGM, RAVGS, C, CM, CS, RD,
     &                   RDM, RDS, LAMB, LAMBA, LAMBB, LAMD, LAMDA,
     &                   LAMDB, MW, MWA, MWB)

            MINPHI = 1.0E+36
            DO 230 M = 1, NSYM
               CALL WEIBGN (BZERO, RAND, PHI)
               MINPHI = MIN (PHI, MINPHI)
  230       CONTINUE
            PHI = MINPHI

            IF (VARY .EQ. 0) PHI = 1.0

            IF (IOUT .EQ. 15) WRITE(8,*) 'PHI = ', PHI

C ** CALCULATE REGION DEPENDENT S/N CURVE PARAMETERS

            FACTR = PHI * KRATIO * Z

            DO 235 L = ZROREG, NUMREG
              TRSBND(L) = FACTR * SBND(L)
              TRBIGK(L) = BIGK(L)
  235       CONTINUE
            TRSBND(0) = SBND(0)

            IF (ZROREG .EQ. 0) CALL KOMO (SZERO, BIGK, MM, NBND,
     &          TRSBND, TRBIGK, FACTR, NUMREG)

            DO 250 L = ZROREG, NUMREG
               LNA(L) = MM(L) * ALOG(TRBIGK(L))
               LPHIM(L) = MM(L) * ALOG(PHI)
               IF (IOUT .EQ. 15) THEN
                  WRITE(8,*) 'L =',L,' MM =',MM(L),' BIGK =',TRBIGK(L)
                  WRITE(8,*) 'LNA =', LNA(L), ' PHI =', PHI
                  WRITE(8,*) 'LPHIM =', LPHIM(L), ' SBND =', SBND(L)
                  WRITE(8,*) 'KRATIO = ', KRATIO, ' Z = ', Z
                  WRITE(8,*) 'TRSBND =', TRSBND(L), ' FACTR = ', FACTR
               ENDIF
  250       CONTINUE

C ** CALL BLDHLF TO OBTAIN BLADE HCF LIFE

            NEWLIF = BLDHLF (RPM, RROOT, RAVG, C, RD, LAMB, LAMD, MW,
     &                       IMIN, MD, NB, NS, A0, A1, B0, B1, C0, C10,
     &                       C11, C20, C21, MM, LNA, LPHIM, KRATIO,
     &                       LNZ, SBND, ZROREG, NUMREG, SZERO)

            IF (IOUT .EQ. 15) WRITE(8,*) 'NEWLIF = ', NEWLIF

            IF (NLIFET .GE. 100) CALL INSORT (NEWLIF, LIFE, NLIFET)

  200    CONTINUE

  150 CONTINUE

      IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES TO FILE LOWLIF

         DO 300 J = 1, (NLIFET / 100)
            WRITE(9,*) J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
  300    CONTINUE

C ** INITIALIZE VARIABLE BLFPOS()
```

```
         DO 325 J = 1, MAXBLF
            BLFPOS(J) = 0
  325    CONTINUE

         FIFTY = 0.50E0

C ** PRINT EMPIRICAL BLIVES

         WRITE(3,925)

         DO 350 J = 1, NBLIFE
            BLFPOS(J) = NINT (BLFPER(J) * FLOAT (NLIFET))
            WRITE(3,926) BLFPER(J), LIFE(BLFPOS(J))
  350    CONTINUE
         WRITE(3,926) FIFTY, LIFE(NLIFET/2)

      ENDIF

C ** CALCULATE NORMAL MEDIAN CURVE IF DESIRED

      IF ((VARY .EQ. 3) .AND. (NMED .EQ. 1)) THEN

         CALL SORTM (ALLM, NUMREG, MCOUNT)

         MID = MCOUNT / 2
         DO 400 L = 1, NUMREG
            MEDM(L) = ALLM(MID,L)
  400    CONTINUE

         CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG,
     &                NBND, BIGK1, BZERO)

      ENDIF


  925 FORMAT(///,2X,'B LIVES:      EMPIRICAL',/)

  926 FORMAT(2X,F7.5,5X,E13.6)


      STOP
      END


C*********************************************************************
C                 SAMPLE 'BLDHCD' INPUT FILE
C*********************************************************************
C 675..................................RANDOM NUMBER SEED
C 0....................................OUTPUT DUMP CONTROLLER
C 1....................................INNER LOOP SIZE
C 20000................................OUTER LOOP SIZE
C 54...................................SYMMETRY NUMBER
C 2....................................UNIFORM S/N VARIATION
C 0....................................NORMAL MEDIAN NOT REQUIRED
C 0....................................MAT. PROC. VAR. NOT REQUIRED
C 3....................................NUMBER OF BLIVES REQUESTED
C 0.0001...............................B.01 LIFE
C 0.001................................B.1 LIFE
C 0.01.................................B1 LIFE
C 26161.   600.........................ROTOR SPEED VARIATION PARAMETERS:
C                                        MEAN, STD.DEV. (NORMAL DIST.)
C 4.700  0.0035........................BLADE ROOT RADIUS MEAN & STD DEV
C 5.117  0.0035........................BLADE AVERAGE RADIUS MEAN & STD DEV
C 0.1303 0.0035........................DISTANCE FROM NEUTRAL AXIS MEAN & STD DEV
C 4.445  0.010.........................DAMPER RADIUS MEAN & STD DEV
C 0.0      0.0.........................UNCERTAINTY IN PERFORMACE BALANCE
C 0.50     1.50........................UNCERT. IN DAMPER COEFFICENT OF FRICTION
C 0.40     0.60........................WALKER EXPONENT m
C 0.0004769............................MINIMUM MOMENT OF INERTIA
C 0.0010733............................DAMPER MASS
C 54...................................NUMBER OF ROTOR BLADES
```

```
C 78..............................NUMBER OF STATOR VANES
C
C            COEFFICIENTS OF RESPONSE SURFACE FUNCTIONS
C FLOW RATE:
C    Fmdot(w) = A + B * w
C         A             B
C    -24.41242623   0.3307822E-02
C
C ENTHALPY CHANGE:
C    Fdeltah(w) = A + B * W
C         A             B
C    -29.65037673   0.6433368E-02
C
C BLADE DAMPER EFFECTIVENESS:
C    IF mrw**2 < A
C         Feff(m, r, w) =  B + C * mrw**2
C    IF mrw**2 > A
C          Feff(m, r, w) =  D + E * mrw**2
C    A      B        C          D              E
C    26     1.0   -0.03750    5.683003E-3    7.429614E-4
C
C 'RT, PWA 1480, 001 DIRECTION'.....MATERIAL DESCRIPTION
C 137000.   142000.   1    8.........YIELD & ULTIMATE STRENGTHS, NDIV, NPTS
C 8    -1.0   1....................# PTS IN DIV, STRESS RATIO, REGION
C 80000.      6800.................S(1) N(1)       RAW
C 80000.     15000.................S(2) N(2)   STRESS-LIFE
C 60000.     27000.................S(3) N(3)       (S/N)
C 60000.     43200.................S(4) N(4)        DATA
C 50000.    139300.................S(5) N(5)       POINTS
C 50000.    545200.................S(6) N(6)      FOR THE
C 50000.    147000.................S(7) N(7)      SPECIFIC
C 35000.   4344800.................S(8) N(8)      MATERIAL
C 0.00..............................VALUE OF So SUPPLIED (PSI)
C 1  0..............................NUMBER OF REGIONS:W/DATA W/O DATA
C 1.0E+36...........................LIFE BOUNDARY FOR REGION 1
C 0.00..............................CONSTRAINT ON COEFF. OF VARIATION
C 0    0.00    0.00...................0 PTS IN RANGE, LOWER BOUND, UPPER BOUND
C 0.0    0.0    0.0...................NORMAL DIST. PRIORS: DELTA, Mo, SIGMA2
C
C**********************************************************************
C                            LIST OF VARIABLES
C**********************************************************************
C
C A0, A1      COEFFICIENTS OF THE FLOW RATE, m-dot, RESPONSE SURFACE
C               (PERFORMANCE BALANCE MODEL)
C ALLM()      2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C B0, B1      COEFFICIENTS OF THE ENTHALPY CHANGE, delta-h, RESPONSE SURFACE
C               (PERFORMANCE BALANCE MODEL)
C BIGK()      1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH
C               REGION
C BIGK1       EQUAL TO BIGK(1) - DUMMY PARAMETER FOR CALLS TO SUBROUTINE
C               EXPCTD
C BLDHLF      REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND HCF LIFE
C               CALCULATION
C BLFPER()    1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C BLFPOS()    1-D ARRAY CONTAINING POSITION IN LIFE() OF EMPIRICAL BLIVES
C BZERO       VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C C           SELECTED DISTANCE FROM NEUTRAL AXIS (in)
C C0, C10, C11, C20, C21
C             COEFFICIENTS OF THE BLADE DAMPER EFFECTIVENESS RESPONSE SURFACE
C CM          MEAN OF DISTANCE FROM NEUTRAL AXIS (in)
C CS          STANDARD DEVIATION OF DISTANCE FROM NEUTRAL AXIS (in)
C FACTR       SCALE FACTOR EQUAL TO PHI * KRATIO * Z
C FIFTY       EQUAL TO .5 - USED TO ACCESS 50% POINT IN LIFE()
C FTU         MATERIAL ULTIMATE STRENGTH (psi)
C FTY         MATERIAL YIELD STRENGTH (psi)
C I           CONTROLS INNER DO LOOP
C IMIN        MINIMUM MOMENT OF INERTIA (in**4)
C IOUT        CONTROLS DUMP TO FILE IOUTPR
C J           CONTROLS DO LOOP FOR EACH BLIFE
C K           CONTROLS OUTER DO LOOP
C KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C L           CONTROLS DO LOOP FOR EACH REGION
C LAMB        SELECTED UNCERTAINTY IN PERFORMACE BALANCE MODEL, LAMbdaB
C LAMBA       UNCERTAINTY IN PERFORMANCE BALANCE MODEL, LAMbdaB, UNIFORM
```

```
C                 DISTRIBUTION LOWER BOUND
C LAMBB        UNCERTAINTY IN PERFORMANCE BALANCE MODEL, LAMbdaB, UNIFORM
C                 DISTRIBUTION UPPER BOUND
C LAMD         SELECTED UNCERTAINTY IN DAMPER COEFFICENT OF FRICTION, LAMbdaD
C LAMDA        UNCERTAINTY IN DAMPER COEFFICIENT OF FRICTION, LAMbdaD, UNIFORM
C                 DISTRIBUTION LOWER BOUND
C LAMDB        UNCERTAINTY IN DAMPER COEFFICIENT OF FRICTION, LAMbdaD, UNIFORM
C                 DISTRIBUTION UPPER BOUND
C LIFE()       1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
C                 - SORTED VALUES OF THE LEFT-HAND TAIL
C LNA()        1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
C LNZ          NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
C LPHIM()      1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
C M            CONTROLS SYMMETRY DO LOOP
C MAXBLF       MAXIMUM NUMBER OF BLIVES TO BE PROVIDED
C MAXDAT       MAXIMUM NUMBER OF POINTS PER DATA SET PER REGION ALLOWED
C MAXLIF       MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA,
C                 ALPHA CALCULATION
C MAXMM        MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION
C MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C MCOUNT       NUMBER OF M's TO BE USED TO CALCULATE THE TRUNCATED NORMAL
C                 MEDIAN S/N CURVE
C MD           DAMPER MASS (lbm)
C MEDM()       1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C MID          POINTER TO THE MEDIAN M VALUES - EQUAL TO HALF OF MCOUNT
C MINPHI       EQUAL TO MIN(PHI) - THE MINIMUM OF NSYM DRAWS OF PHI
C MM()         1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C MPROC        Materials PROCess variation - CONTROLS MATERIALS PROCESS
C                 VARIATION - 0 - NO VARIATION;  1 - VARIATION
C MU()         1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
C                 MEAN FOR EACH REGION
C MW           SELECTED WALKER M
C MWA          WALKER M UNIFORM DISTRIBUTION LOWER BOUND
C MWB          WALKER M UNIFORM DISTRIBUTION UPPER BOUND
C NB           NUMBER OF ROTOR BLADES
C NBLIFE       NUMBER OF BLIVES TO BE PROVIDED
C NBND()       1-D ARRAY CONTAINING UPPER BOUNDS FOR THE NUMREG LIFE REGIONS OF
C                 INTEREST FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
C NEWLIF       LIFE VALUE RETURNED FROM CALL TO BLDHLF
C NF()         2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL S/N DATA
C                 SET BROKEN INTO LIFE REGIONS
C NHYPER       SIZE OF OUTER LOOP
C NLIFE        SIZE OF INNER LOOP
C NLIFET       TOTAL NUMBER OF LIVES CALCULATED BY PFM
C NMED         CONTROLS MEDIAN CALCULATION FOR THE TRUNCATED NORMAL
C                 DISTRIBUTION CASE - 0 - NO MEDIAN CALCULATION;  1 - MEDIAN
C                 CALCULATION DESIRED
C NPTS()       1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION FOR
C                 THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
C NS           NUMBER OF STATOR BLADES
C NSYM         SYMMETRY NUMBER
C NUMREG       NUMBER OF REGIONS OF INTEREST
C PHI          WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C PSIG         EQUAL TO SQRT(PVAR) - MATERIALS PROCESS STANDARD DEVIATION
C PVAR         MATERIALS PROCESS VARIATION
C RAND         RANDOM NUMBER SEED
C RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR
C                 EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L)
C                 IS THE UPPER BOUND
C RAVG         SELECTED BLADE AVERAGE RADIUS (in)
C RAVGM        MEAN OF AVERAGE BLADE RADIUS (in)
C RAVGS        STANDARD DEVIATION OF AVERAGE BLADE RADIUS (in)
C RD           SELECTED DAMPER RADIUS (in)
C RDM          MEAN OF DAMPER RADIUS (in)
C RDS          STANDARD DEVIATION OF DAMPER RADIUS (in)
C RPM          SELECTED ROTOR SPEED (rpm)
C RPMM         MEAN OF ROTOR SPEED (rpm)
C RPMS         STANDARD DEVIATION OF ROTOR SPEED (rpm)
C RROOT        SELECTED BLADE ROOT RADIUS (in)
C RROOTM       MEAN OF BLADE ROOT RADIUS (in)
C RROOTS       STANDARD DEVIATION OF BLADE ROOT RADIUS (in)
C SBND()       1-D ARRAY CONTAINING THE STRESS VALUES (psi, R = -1.0)
C                 CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C                 REGION CONTAINED IN NBND()
C SIG()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
```

```
C              DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C STR()       2-D ARRAY CONTAINING STRESS POINTS (STRESS RATIO = -1.0) FOR
C              THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS
C SZERO       STRESS TENSILE TEST POINT, So
C TRBIGK()    1-D ARRAY CONTAINING VALUES OF BIGK() CORRECTED FOR SZERO,
C              PHI, KRATIO, AND Z
C TRSBND()    1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR EACH
C              REGION CALCULATED FOR EACH TRIAL
C VARY        CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;
C              1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;  3 -
C              TRUNCATED NORMAL VARIATION
C Z           LOGNORMAL(0,PVAR) GENERATED RANDOM VARIATE
C ZROREG      ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              BEGINNING VALUE -- 0 ZERO REGION EXISTS, 1 - NO ZERO REGION


C***********************************************************************


C  SUBROUTINE DRVRIN READS AND ECHOES THE INPUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  31OCT90      COMMENTS:  20APR92
C     VERSION:  BLDHCF V1, V1.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE DRVRIN (RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS,
     &                   CM, CS, RDM, RDS, LAMBA, LAMBB, LAMDA,
     &                   LAMDB, MWA, MWB, IMIN, MD, NB, NS, A0, A1,
     &                   B0, B1, C0, C10, C11, C20, C21)

C  OUTPUT:  RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS, CM, CS, RDM, RDS,
C           LAMBA, LAMBB, LAMDA, LAMDB, MWA, MWB, IMIN, MD, NB, NS,
C           A0, A1, B0, B1, C0, C10, C11, C20, C21

C     IMPLICIT NONE

      COMMON   IOUT

      INTEGER  IOUT, NB, NS

      REAL     A0, A1, B0, B1, C0, C10, C11, C20, C21, CM, CS, IMIN,
     &         LAMBA, LAMBB, LAMDA, LAMDB, MD, MWA, MWB, RAVGM, RAVGS,
     &         RDM, RDS, RPMM, RPMS, RROOTM, RROOTS


C                    LIST OF VARIABLES
C
C  A0, A1   Coefficients of the flow rate, m-dot, response surface
C             (performance balance model)
C  B0, B1   Coefficients of the enthalpy change, delta-h, response surface
C             (performance balance model)
C  C0, C10, C11, C20, C21
C           Coefficients of the blade damper effectiveness response surface
C  CM       Mean of distance from neutral axis (in)
C  CS       Standard deviation of distance from neutral axis (in)
C  IMIN     Minimum moment of inertia (in**4)
C  IOUT     Output dump controller
C  LAMBA    Uncertainty in performance balance model, LAMbdaB, Uniform
C             distribution lower bound
C  LAMBB    Uncertainty in performance balance model, LAMbdaB, Uniform
C             distribution upper bound
C  LAMDA    Uncertainty in damper coefficient of friction, LAMbdaD, Uniform
C             distribution lower bound
C  LAMDB    Uncertainty in damper coefficient of friction, LAMbdaD, Uniform
C             distribution upper bound
C  MD       Damper mass (lbm)
C  MWA      Walker m Uniform distribution lower bound
C  MWB      Walker m Uniform distribution upper bound
C  NB       Number of rotor blades
C  NS       Number of stator vanes
```

```
C   RAVGM    Mean of average blade radius (in)
C   RAVGS    Standard deviation of average blade radius (in)
C   RDM      Mean of damper radius (in)
C   RDS      Standard deviation of damper radius (in)
C   RPMM     Mean of rotor speed (rpm)
C   RPMS     Standard deviation of rotor speed (rpm)
C   RROOTM   Mean of blade root radius (in)
C   RROOTS   Standard deviation of blade root radius (in)


      READ(1,*) RPMM,  RPMS,   RROOTM, RROOTS, RAVGM, RAVGS,
     &          CM,    CS,     RDM,    RDS,
     &          LAMBA, LAMBB, LAMDA,   LAMDB,   MWA,   MWB,
     &          IMIN, MD, NB, NS,
     &          A0, A1, B0, B1, C0, C10, C11, C20, C21

      WRITE(3,900)
      WRITE(3,901) RPMM,  RPMS,   RROOTM, RROOTS, RAVGM, RAVGS,
     &             CM,    CS,     RDM,    RDS
      WRITE(3,902) LAMBA, LAMBB, LAMDA,   LAMDB,  MWA,   MWB
      WRITE(3,903) IMIN, MD, NB, NS
      WRITE(3,904) A0, A1, B0, B1, C0, C10, C11, C0, C20, C21

  900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &        'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &        'NASA Contract NAS7-918 is acknowledged.',////,
     &        33X,'INPUT DATA',///,14X,'DRIVERS',31X,'DISTRIBUTIONS',
     &        //,49X,'N( MEAN,   STD. DEV.)')

  901 FORMAT(//,2X,'ROTOR SPEED VARIATION (rpm)',20X,
     &        'N(',F8.1,',',F8.1,')',
     &        //,2X,'BLADE ROOT RADIUS (in)',25X,
     &        'N(',F6.3,',',E10.3,')',
     &        //,2X,'BLADE AVERATE RADIUS (in)',22X,
     &        'N(',F6.3,',',E10.3,')',
     &        //,2X,'DISTANCE FROM NEUTRAL AXIS (in)',16X,
     &        'N(',F7.4,',',E9.2,')',
     &        //,2X,'DAMPER RADIUS (in)',29X,'N(',F6.3,',',E10.3,')')

  902 FORMAT(//,2X,'UNCERTAINTY IN PERFORMANCE BALANCE',14X,
     &        'U(',F7.4,',',F8.4,')',
     &        //,2X,'DAMPER COEFFICIENT OF FRICTION',18X,
     &        'U(',F7.4,',',F8.4,')',
     &        //,2X,'WALKER m',40X,'U(',F7.4,',',F8.4,')')

  903 FORMAT(////,23X,'OTHER GEOMETRIC INPUT',
     &        ///,2X,'MINIMUM MOMENT OF INERTIA (in**4)',19X,E10.4,
     &        //,2X,'DAMPER MASS (lb)',36X,E11.5,
     &        //,2X,'NUMBER OF ROTOR BLADES',29X,I2,
     &        //,2X,'NUMBER OF STATOR VANES',29X,I2)

  904 FORMAT(////,13X,'COEFFICIENTS OF RESPONSE SURFACE FUNCTIONS',
     &        //,2X,'FLOW RATE:',
     &        /,5X,'Fmdot(w) = ',F12.8,' + ',E14.7,' * w',
     &        //,2X,'ENTHALPY CHANGE:',
     &        /,5X,'Fdeltah(w) = ',F12.8,' + ',E14.7,' * w',
     &        //,2X,'BLADE DAMPER EFFECTIVENESS:',
     &        /,5X,'IF mrw**2 < ',F4.1,
     &        /,10X,'Feff(m, r, w) = ',E14.7,' + ',E14.7,' * mrw**2',
     &        /,5X,'IF mrw**2 > ',F4.1,
     &        /,10X,'Feff(m, r, w) = ',E14.7,' + ',E14.7,' * mrw**2')

      RETURN
      END


C***********************************************************************


C  SUBROUTINE SELECT PERFORMS THE DRIVER SELECTION
C  PROGRAMMER:  L. NEWLIN
C         DATE:  31OCT90      COMMENTS:  20APR92
```

```
C     VERSION:  BLDHCF V1, V1.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE SELECT (RAND, RPM, RPMM, RPMS, RROOT, RROOTM,
     &                   RROOTS, RAVG, RAVGM, RAVGS, C, CM, CS, RD,
     &                   RDM, RDS, LAMB, LAMBA, LAMBB, LAMD, LAMDA,
     &                   LAMDB, MW, MWA, MWB)

C        INPUT:  RAND, RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS, CM,
C                CS, RDM, RDS, LAMBA, LAMBB, LAMDA, LAMDB, MWA, MWB
C       OUTPUT:  RPM, RROOT, RAVG, C, RD, LAMB, LAMD, MW
C SUBPROGRAMS:  NORMGN, PRYRV

C     IMPLICIT NONE

      COMMON   IOUT

      INTEGER  IOUT

      REAL     C, CM, CS, DUM, LAMB, LAMBA, LAMBB, LAMD, LAMDA, LAMDB,
     &         MW, MWA, MWB, RAVG, RAVGM, RAVGS, RD, RDM, RDS, RPM,
     &         RPMM, RPMS, RROOT, RROOTM, RROOTS

      DOUBLE PRECISION  RAND


C                     LIST OF VARIABLES
C
C  C       Selected distance from neutral axis (in)
C  CM      Mean of distance from neutral axis (in)
C  CS      Standard deviation of distance from neutral axis (in)
C  DUM     Dummy variable
C  IOUT    Output dump controller
C  LAMB    Selected uncertainty in performace balance model, LAMbdaB
C  LAMBA   Uncertainty in performance balance model, LAMbdaB, Uniform
C            distribution lower bound
C  LAMBB   Uncertainty in performance balance model, LAMbdaB, Uniform
C            distribution upper bound
C  LAMD    Selected uncertainty in damper coefficent of friction, LAMbdaD
C  LAMDA   Uncertainty in damper coefficient of friction, LAMbdaD, Uniform
C            distribution lower bound
C  LAMDB   Uncertainty in damper coefficient of friction, LAMbdaD, Uniform
C            distribution upper bound
C  MW      Selected Walker m
C  MWA     Walker m Uniform distribution lower bound
C  MWB     Walker m Uniform distribution upper bound
C  RAND    Random number seed
C  RAVG    Selected blade average radius (in)
C  RAVGM   Mean of average blade radius (in)
C  RAVGS   Standard deviation of average blade radius (in)
C  RD      Selected damper radius (in)
C  RDM     Mean of damper radius (in)
C  RDS     Standard deviation of damper radius (in)
C  RPM     Selected rotor speed (rpm)
C  RPMM    Mean of rotor speed (rpm)
C  RPMS    Standard deviation of rotor speed (rpm)
C  RROOT   Selected blade root radius (in)
C  RROOTM  Mean of blade root radius (in)
C  RROOTS  Standard deviation of blade root radius (in)


      CALL NORMGN (RAND, RPMM,   RPMS,   RPM)
      CALL NORMGN (RAND, RROOTM, RROOTS, RROOT)
      CALL NORMGN (RAND, RAVGM,  RAVGS,  RAVG)
      CALL NORMGN (RAND, CM,     CS,     C)
      CALL NORMGN (RAND, RDM,    RDS,    RD)

      CALL PRYRV (RAND, LAMBA, LAMBB, LAMDA, LAMDB, LAMB, LAMD)
      CALL PRYRV (RAND, MWA,   MWB,   MWA,   MWB,   MW,   DUM)
```

```
      IF (IOUT .EQ. 15) THEN
         WRITE(8,*) 'RPM = ', RPM, ' RROOT = ', RROOT
         WRITE(8,*) 'RAVG = ', RAVG, ' C = ', C
         WRITE(8,*) 'RD = ', RD, ' MW = ', MW
         WRITE(8,*) 'LAMB = ', LAMB, ' LAMD = ', LAMD
      ENDIF

      RETURN
      END


C***********************************************************************


C  FUNCTION BLDHLF PERFORMS/CONTROLS THE DRIVER TRANSFORMATION AND LIFE
C  CALCULATION FOR THE BLADE HCF MODEL
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20APR92
C     VERSION:  BLDHCF V1.1 (MATCHR V8.5)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      FUNCTION BLDHLF (RPM, RROOT, RAVG, C, RD, LAMB, LAMD, MW, IMIN,
     &                 MD, NB, NS, A0, A1, B0, B1, C0, C10, C11, C20,
     &                 C21, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG,
     &                 NUMREG, SZERO)

C         INPUT:  RPM, RROOT, RAVG, C, RD, LAMB, LAMD, MW, IMIN, MD, NB, NS,
C                 A0, A1, B0, B1, C0, C10, C11, C20, C21, MM, LNA, LPHIM,
C                 KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO
C        OUTPUT:  BLDHLF
C  SUBPROGRAMS:   GTLIFE

C     IMPLICIT NONE

      COMMON   IOUT

      INTEGER  IOUT, MAXREG, NB, NS, NUMREG, ZROREG

      PARAMETER (MAXREG = 3)

      REAL     A0, A1, B0, B1, BLDHLF, C, C0, C10, C11, C20, C21,
     &         DELTAH, GTLIFE, IMIN, KRATIO, LAMB, LAMD, LIFE,
     &         LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MD, MDOT,
     &         MM(0:MAXREG), MRW2, MW, NF, R, RAVG, RD, RPM, RROOT,
     &         SALT, SBND(0:MAXREG), SBRM, SDSUD, SEQ, SMAX, SMEAN,
     &         SMIN, SUD, SZERO


C                      LIST OF VARIABLES
C
C  A0, A1   Coefficients of the flow rate, m-dot, response surface
C              (performance balance model)
C  B0, B1   Coefficients of the enthalpy change, delta-h, response surface
C              (performance balance model)
C  BLDHLF   Real function performing the driver transformation and HCF life
C              calculation
C  C        Selected distance from neutral axis (in)
C  C0, C10, C11, C20, C21
C           Coefficients of the blade damper effectiveness response surface
C  DELTAH   Enthalpy change, delta-h
C  GTLIFE   Function which calculates the cycles to failure at a given stress
C  IMIN     Minimum moment of inertia (in**4)
C  IOUT     Output dump controller
C  KRATIO   Ratio of K*/K, constant over regions and components
C  LAMB     Selected uncertainty in performace balance model, LAMbdaB
C  LAMD     Selected uncertainty in damper coefficent of friction, LAMbdaD
C  LIFE     Fatigue life in seconds
```

```
C  LNA()    1-D array containing ln(A) = ln(BIGK) * MM for each region
C  LNZ      Normal(0,PVAR) generated random variate
C  LPHIM()  1-D array containing ln(PHI) * MM for each region
C  MAXREG   Maximum number of regions allowed
C  MD       Damper mass (lbm)
C  MDOT     Flow rate, m-dot
C  MM()     1-D array containing selected values of m for each region
C  MRW2     Damper normal load (lbf)
C  MW       Selected Walker m
C  NB       Number of rotor blades
C  NF       Fatigue life in cycles
C  NUMREG   Number of regions of interest
C  NS       Number of stator vanes
C  R        Stress ratio
C  RAVG     Selected blade average radius (in)
C  RD       Selected damper radius (in)
C  RPM      Selected rotor speed (rpm)
C  RROOT    Selected blade root radius (in)
C  SALT     Alternating stress (psi)
C  SBND()   1-D array containing the stress values (psi, R = -1.0)
C             corresponding to the "life boundary" values for each region
C             contained in NBND()
C  SBRM     Blade root mean stress (psi)
C  SDSUD    Ratio of damped to undamped vibratory stress
C  SEQ      Equivalent zero mean stress (psi)
C  SMAX     Maximum or peak stress (psi)
C  SMEAN    Mean stress (psi)
C  SMIN     Minimum stress (psi)
C  SUD      Blade undamped vibratory stress (psi)
C  SZERO    Stess tensile test point, So
C  ZROREG   ZeRO REGion - values chosen to facilitate region DO loop
C             beginning value - 0 - zero region exists, 1 - no zero region


      IF (IOUT .EQ. 15) THEN
         WRITE(8,*) 'RPM = ', RPM, ' RROOT = ', RROOT
         WRITE(8,*) 'RAVG = ', RAVG, ' C = ', C
         WRITE(8,*) 'RD = ', RD, ' MW = ', MW
         WRITE(8,*) 'LAMB = ', LAMB, ' LAMD = ', LAMD
         WRITE(8,*) 'IMIN = ', IMIN, ' MD = ', MD
         WRITE(8,*) 'NB = ', NB, ' NS = ', NS
      ENDIF

C     CALCULATE FLOW CONDITIONS

      MDOT   = LAMB * (A0 + A1 * RPM)
      DELTAH = LAMB * (B0 + B1 * RPM)

C     CALCULATE BLADE ROOT MEAN STRESS

      SBRM = (MDOT * DELTAH / RPM) * (C / (IMIN * FLOAT (NB)))
     &       * (1.0 - (RROOT / RAVG)) * 9336

C     OBTAIN BLADE UNDAMPED VIBRATORY STRESS

      SUD = (8.55300181 + 34.06551173 * (SBRM / 9336)) * 1000.0

C     CALCULATE DAMPER NORMAL LOAD

      MRW2 = (MD * RD * (RPM ** 2)) * 2.83805E-5

C     OBTAIN BLADE DAMPER EFFECTIVENESS - THE RATIO OF THE DAMPED TO
C     UNDAMPED VIBRATORY STRESS

      IF (MRW2 .LT. C0) THEN
         SDSUD = LAMD * (C10 + C11 * MRW2)
      ELSE
         SDSUD = LAMD * (C20 + C21 * MRW2)
      ENDIF

C     CALCULATE ALTERNATING & MEAN STRESSES, MAX & MIN STRESSES,
C     AND THE STRESS RATIO

      SALT = SUD * (SDSUD)
```

```
      SMEAN = SBRM
      SMAX = SMEAN + SALT
      SMIN = SMEAN - SALT
      R = SMIN / SMAX

C     CALCULATE EQUIVALENT ZERO MEAN STRESS USING WALKER RELATION

      SEQ = SMAX * ((1.0 - R) / 2.0) ** MW

C     OBTAIN FATIGUE LIFE (IN CYCLES) FROM MATERIALS MODEL

      NF = GTLIFE (SEQ, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG,
     &             NUMREG, SZERO)

C     TRANSFORM LIFE FROM CYCLES TO SECONDS

      LIFE = (60.0 / RPM) * NF / FLOAT (NS)

      BLDHLF = LIFE

      IF (IOUT .EQ. 15) THEN
         WRITE(8,*) 'MDOT = ', MDOT, ' DELTAH = ', DELTAH
         WRITE(8,*) 'SBRM = ', SBRM, ' SUD = ', SUD
         WRITE(8,*) 'MRW2 = ', MRW2, ' SDSUD = ', SDSUD
         WRITE(8,*) 'SALT = ', SALT, ' SMEAN = ', SMEAN
         WRITE(8,*) 'SMAX = ', SMAX, ' SMIN = ', SMIN
         WRITE(8,*) 'R = ', R, ' SEQ = ', SEQ
         WRITE(8,*) 'NF = ', NF, ' LIFE = ', LIFE
      ENDIF


      RETURN
      END


C***********************************************************************

C  SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20JUL90
C     VERSION:  2.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

C  INPUTS:   NEWLIF, LIFE, NLIFET
C  OUTPUTS:  LIFE

C     IMPLICIT NONE

      INTEGER  MAXLIF

      PARAMETER (MAXLIF = 10000)

      COMMON   IOUT

      INTEGER  I, IOUT, NLIFET, NUM, PLACE

      REAL     LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

C                 LIST OF VARIABLES
C
C  I        CONTROLS DO LOOP FOR INSERTION
C  IOUT     OUTPUT DUMP CONTROLLER
C  LIFE()   1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
C             PFM TO BE SORTED
C  MAXLIF   MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
C             CALCULATION
C  NEWLIF   LIFE VALUE TO BE INSERTED INTO LIFE()
```

```
C  NLIFET    TOTAL NUMBER OF LIVES CALCULATED BY PFM
C  NUM       NUMBER OF LIFE VALUES IN LIFE()
C  PLACE     POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
C  TEMP()    1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON
C              INSERTION OF NEWLIF


      NUM = NLIFET / 2

C     FIND POSITION IN LIFE() FOR NEWLIF

      IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

      DO 100 I = 1, NUM
         IF (NEWLIF .LT. LIFE(I)) THEN
            PLACE = I
            GOTO 110
         ENDIF
  100 CONTINUE
  110 CONTINUE

C     STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()

      DO 200 I = (PLACE + 1), NUM
         TEMP(I) = LIFE(I-1)
  200 CONTINUE

C     INSERT NEWLIF

      LIFE(PLACE) = NEWLIF

C     SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

      DO 300 I = (PLACE + 1), NUM
         LIFE(I) = TEMP(I)
  300 CONTINUE

C     IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN

  400 CONTINUE


      RETURN
      END

C***********************************************************************
C   SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C   INDEPENDENT RANDOM VARIATES
C   PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C         DATE:  9MAR87
C   SUBPROGRAM:  RANDOM
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, RHO1, RHO2, THE1, THE2, X, Y

      INTEGER IOUT

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      X = FRAC * (RHO2 - RHO1) + RHO1

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
```

```
      Y = FRAC * (THE2 - THE1) + THE1

      IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ' RHO2 =', RHO2,
     &       ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y

      RETURN
      END

C***********************************************************************


C   SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
C   AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C   FOR THE STRESS FORMULATION
C   PROGRAMMER:  L. NEWLIN
C         DATE:  13JUL89      FORMAT/COMMENTS:  12AUG91
C      VERSION:  MATCHR V8.4, V8.5   MATGRM V4.4, V4.5
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.


      SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG,
     &                   NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC,
     &                   KRATIO, PVAR)

C   INPUTS:     READS DATA FROM SPECFD AND RELATD; VARY, MPROC
C   OUTPUTS:    RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG, NUMREG,
C               NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
C   SUBPROGRAMS:  INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDREG,
C                 CONCAV, MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR
C         FILES:  5:RELATD-OLD; 6:RELATO-NEW

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
     &        NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
     &        NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG

      REAL    BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
     &        FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
     &        KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
     &        MCHAT(2, MAXREG), MEDM(MAXREG), MO(MAXREG), MU(MAXREG),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        PVAR, RANGEM(2, MAXREG), RATSTR(MAXDAT, 0:MAXSET),
     &        RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
     &        SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT, MAXREG),
     &        SUHAT2(MAXREG), SWHAT2(MAXREG), SX2(MAXREG),
     &        SXY(MAXREG), SY2(MAXREG), SZERO


C                      LIST OF VARIABLES
C
C  BIGKHT        EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
C  BZERO         VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE S/N
C                  DATA SET
C  CZERO         EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                  COEFFICIENT OF VARIATION, Co
C  DD()          1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()       1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
C                  AND SIG() CALCULATION
C  FTUZ          ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ          YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  IOUT          OUTPUT DUMP CONTROLLER
C  IZERO()       2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
```

```
C                 FOR EACH REGION
C   JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C   KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C   L           CONTROLS DO LOOP FOR EACH REGION
C   LAMN        LAMBDA-N - RATIO OF Var(Ln N given S) / (m**2 C**2),
C                 CONSTANT OVER REGIONS AND COMPONENTS
C   LNNF()      3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C   LNSTR()     3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C   MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C   MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                 REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                 - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                 BOUND
C   MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                 MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   MCPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MC() FOR EACH REGION
C   MEDM()      1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C   MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 MEAN FOR EACH REGION
C   MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C   MPROC       Materials PROCess variation -CONTROLS MATERIALS PROCESS
C                 VARIATION - 0 - NO VARIATION;  1 - VARIATION
C   MU()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION MEAN FOR EACH REGION
C   MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C   NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST
C   NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                 SET IN EACH REGION
C   NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                 ALL DATA SETS IN A REGION (Number of Points Per Region)
C   NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C   NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   PVAR        MATERIALS PROCESS VARIATION
C   RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C   RATSTR()    2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
C                 STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C   RAWNF()     2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                 DATA SETS
C   RAWSTR()    2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
C                 DATA (%) FOR ALL S/N DATA SETS
C   REFNP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C   SIG()       1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C   SIGMA2()    1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 VARIANCE FOR EACH REGION
C   STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SUHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SWHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SX2()       1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                 (X = Ln S)
C   SXY()       1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR EACH
C                 REGION (X = Ln S, Y = Ln N)
C   SY2()       1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                 (Y = Ln N)
C   SZERO       STRESS TENSILE TEST POINT, So
```

```
C  VARY          CONTROLS TYPE OF CURVE VARIATION DESIRED — 0 - NO
C                  VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                  VARIATION;  3 - TRUNCATED NORMAL VARIATION
C  ZROREG        ZeRO REGion — VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION


      OPEN(5, FILE = 'RELATD', STATUS = 'OLD')
      OPEN(6, FILE = 'RELATO', STATUS = 'NEW')

C  RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C  RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET
C  INFORMATION

C  PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION

C  INITIALIZE PRIMARY ARRAYS

      CALL INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
     &           NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

C  READ, CONVERT, ECHO INFORMATION

      CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR,
     &          LNNF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, NNODAT,
     &          NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO,
     &          SIGMA2, KRATIO, LAMN)

C  CALCULATE RESIDUAL VARIANCES

      CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD,
     &             SWHAT2, SUHAT2, NPPR)

C  CALCULATE M CONTRAINT BASED ON Co

      CALL FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)


      IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1) .OR. (VARY .EQ. 2)) THEN

C  CALCULATIONS FOR ALL TYPES OF VARIATION SAVE NORMAL

C  CALCULATE BOUNDS FOR CONFIDENCE INTERVALS

         CALL INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                JZERO, MCHAT)

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C  OBTAIN POSTERIOR RANGES ON M

         CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT,
     &                RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

         CALL ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

         CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

         WRITE(7,900)

         DO 25 L = 1, NUMREG
            WRITE(7,905) L, IZERO(1, L), IZERO(2, L),
     &                      JZERO(1, L), JZERO(2, L)
```

```
 25     CONTINUE

        WRITE(7,910)

        DO 50 L = 1, NUMREG
           WRITE(7,915) L, MCHAT(2,L), MCHAT(1,L)
 50     CONTINUE

        IF (CZERO .GT. 0.0) THEN
           WRITE(7,960)
           DO 150 L = 1, NUMREG
              IF (MCPNT(L) .EQ. 1) THEN
                 WRITE(7,965) L, MC(1,L)
              ELSEIF (MCPNT(L) .EQ. 2) THEN
                 WRITE(7,970) L, MC(1,L), MC(2,L)
              ENDIF
150        CONTINUE
        ENDIF

        WRITE(7,920)
        WRITE(7,930)

        DO 100 L = 1, NUMREG
           WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
100     CONTINUE

        WRITE(7,950)

C  CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO

        CALL MEDIAN (NUMREG, RANGEM, MEDM)

C  CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS

        CALL EXPCTD (1, MEDM, REFNP, STR, NF, SZERO, NUMREG, ZROREG,
     &               NBND, BIGKHT, BZERO)

C  CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED

        IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
           DO 200 L = 1, NUMREG
              RANGEM(1,L) = MEDM(L)
              RANGEM(2,L) = MEDM(L)
200        CONTINUE
        ENDIF

     ELSE

C  NORMAL VARIATION IS DESIRED

C  CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION

        CALL MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO,
     &              SIGMA2, MCHAT, MU, SIG)

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

     IF (MPROC .EQ. 1) THEN
        CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
     ENDIF

C  COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M

        CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

        CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO,
     &               MPNT, MO, SIGMA2)

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

        CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP
```

```
      WRITE(7,975)

      DO 350 L = 1, NUMREG
         WRITE(7,980) L, MCHAT(1,L)
 350  CONTINUE

      IF (CZERO .GT. 0.0) THEN
         WRITE(7,960)
         DO 360 L = 1, NUMREG
            IF (MCPNT(L) .EQ. 1) THEN
               WRITE(7,965) L, MC(1,L)
            ELSEIF (MCPNT(L) .EQ. 2) THEN
               WRITE(7,970) L, MC(1,L), MC(2,L)
            ENDIF
 360     CONTINUE
      ENDIF

      WRITE(7,920)
      WRITE(7,930)

      DO 370 L = 1, NUMREG
         WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
 370  CONTINUE

      WRITE(7,950)

      WRITE(7,985)
      DO 380 L = 1, NUMREG
         WRITE(7,990) L, MU(L), SIG(L)
 380 CONTINUE

    ENDIF

C  PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS

    IF (MPROC .EQ. 1) THEN
       WRITE(7,995) PVAR
    ENDIF


C  FORMAT STATEMENTS

 900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
   &        'Technology.  U.S. Government',/,2X,'Sponsorship under ',
   &        'NASA Contract NAS7-918 is acknowledged.',////,
   &        2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
   &        ///,2X,'95% CONFIDENCE INTERVALS ON C AND m ',
   &        'FOR EACH REGION',/)

 905 FORMAT(7X,'REGION:  ',I1,7X,'Io = (',F12.9,',',F12.9,')',
   &        /,24X,'Jo = (',F12.9,',',F12.9,')')

 910 FORMAT(//,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
   &        //,7X,'REGION',8X,'E(C)',12X,'E(m)',/)

 915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)

 920 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON m FOR EACH '
   &        'REGION')

 930 FORMAT(//,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

 940 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

 950 FORMAT(///)

 960 FORMAT(//,2X,'RANGE ON m FOR EACH REGION IMPLIED BY C '
   &        'CONSTRAINT',
   &        //,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

 965 FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')

 970 FORMAT(6X,I1,8X,F8.4,8X,F8.4)
```

```
  975 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &        'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &        'NASA Contract NAS7-918 is acknowledged.',////,
     &        2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
     &        ///,2X,'ESTIMATE OF m FOR EACH REGION',
     &        //,7X,'REGION',12X,'E(m)',/)

  980 FORMAT(9X,I1,11X,F10.6)

  985 FORMAT(2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS',
     &         //,2X,'REGION',5X,'MEAN',8X,'STD DEV',/)

  990 FORMAT(5X,I1,5X,F7.4,5X,E11.5)

  995 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ',
     &        'MEDIAN S/N CURVE',/,2X,'WARRANTED BY THE AVAILABLE ',
     &        'INFORMATION',//,7X,E11.5)

      RETURN
      END


C***********************************************************************


C   SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C   ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C   PROGRAMMER:  L. NEWLIN
C         DATE:  5OCT87
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRMNAT

      WRITE(8,*) 'PROGRAM EXECUTION TERMINATED'
      STOP
      END


C***********************************************************************


C   SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C   USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  21JUN88       COMMENTS:  13JUL89
C      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR,
     &                 REFNP, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

C   INPUTS:     —
C   OUTPUTS:    NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
C               NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
     &        NPTS(0:MAXSET), REFNP(MAXREG)

      REAL    DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
```

```
     &          MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
     &          RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &          RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &          STR(MAXDAT, MAXREG)
C
C                      LIST OF VARIABLES
C
C   DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                 SIG() CALCULATION
C   I           CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C   IOUT        OUTPUT DUMP CONTROLLER
C   J           CONTROLS DO LOOP FOR EACH DATA SET
C   K           CONTROLS DO LOOP FOR EACH POINT IN A REGION
C   L           CONTROLS DO LOOP FOR EACH REGION
C   LNNF()      3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C   LNSTR()     3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C   MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C   MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 MEAN FOR EACH REGION
C   MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C   MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C   NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                 IN EACH REGION
C   NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C   RATSTR()    2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
C                 STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C   RAWNF()     2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                 DATA SETS
C   RAWSTR()    2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OF TOTAL STRAIN
C                 DATA (%) FOR ALL S/N DATA SETS
C   REFNP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C   SIGMA2()    1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 VARIANCE FOR EACH REGION
C   STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

      DO 100 J = 0, MAXSET
         NPTS(J) = 0.0
  100 CONTINUE

      DO 200 L = 1, MAXREG
         DO 250 J = 0, MAXSET
            NP(J, L) = 0.0
  250    CONTINUE
  200 CONTINUE

      DO 300 J = 0, MAXSET
         DO 350 I = 1, MAXDAT
            RAWNF(I,J) = 0.0
            RAWSTR(I,J) = 0.0
            RATSTR(I,J) = 0.0
  350    CONTINUE
  300 CONTINUE

      DO 400 L = 1, MAXREG
         DO 425 K = 1, MAXDAT
            DO 450 J = 0, MAXSET
               LNNF(K,J,L) = 0.0
               LNSTR(K,J,L) = 0.0
  450       CONTINUE
  425    CONTINUE
  400 CONTINUE

      DO 500 L = 1, MAXREG
         DO 550 K = 1, MAXDAT
          NF(K,L) = 0.0
          STR(K,L) = 0.0
```

```
550     CONTINUE
500 CONTINUE

      DO 600 L = 1, MAXREG
         REFNP(L) = 0
         MPNT(L) = 0
         MZERO(1,L) = 0.0
         MZERO(2,L) = 0.0
         DELTA(L) = 0.0
         MO(L) = 0.0
         SIGMA2(L) = 0.0
600 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE RCE "READS" THE DATA FROM SPECFD AND RELATD; "CONVERTS"
C  THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO
C  SPECFO AND RELATO.  RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS
C  SPECIFIED BY USER
C  PROGRAMMER:  L. NEWLIN
C        DATE:  21JUN88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP,
     &                LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
     &                NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
     &                FTUZ, FTYZ, DELTA, MO, SIGMA2, KRATIO, LAMN)

C  INPUTS:    VARY, MPROC
C  OUTPUTS:   NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP,
C             STR, NF, SZERO, ZROREG, NUMREG, NNODAT, NSETS, NBND,
C             CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2,
C             KRATIO, LAMN
C  SUBPROGRAMS:  TRMNAT, CONVRT

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON IOUT

      INTEGER COUNT, I, IOUT, J, K, L, M, MPNT(MAXREG), MPROC, NDIV,
     &        NNODAT, NP(0:MAXSET, MAXREG), NPTS(0:MAXSET), NSETS,
     &        NUM, NUMREG, REFNP(MAXREG), REG, VARY, ZROREG

      REAL    CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTYZ,
     &        KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        RATIO, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG), SZERO

      CHARACTER*40  DESCRP(0:MAXSET)


C                         LIST OF VARIABLES
C
C  COUNT       INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO,
C                CONVERSION, AND BREAK UP
C  CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                SIG() CALCULATION
C  DESCRP()    1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
```

```
C  FTU          ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
C  FTUZ         ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTY          YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ         YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  I            CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT         OUTPUT DUMP CONTROLLER
C  J            CONTROLS DO LOOP FOR EACH DATA SET
C  K            CONTROLS DO LOOP FOR EACH POINT IN A REGION
C  KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LAMN         LAMBDA-N - RATIO OF Var (Ln N given S) / (m**2 C**2),
C                 CONSTANT OVER ALL REGIONS AND COMPONENTS
C  LNNF()       3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()      3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  M            CONTROLS DO LOOP FOR EACH DATA DIVISION
C  MAXDAT       MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET       MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MO()         1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 MEAN FOR EACH REGION
C  MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MPROC        Materials PROcess variation - CONTROLS MATERIALS PROCESS
C                 VARIATION - 0 - NO VARIATION;  1 - VARIATION
C  MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  NBND()       1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST
C  NDIV         NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO,
C                 REGION PAIRS DURING INPUT
C  NF()         2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NNODAT       Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NP()         2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                 IN EACH REGION
C  NPTS()       1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUM          NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RATIO        STRESS RATIO (R = -1.0 IS DESIRED)
C  RATSTR()     2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
C                 RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C  RAWNF()      2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                 DATA SETS
C  RAWSTR()     2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
C                 DATA (%) FOR ALL S/N DATA SETS
C  REFNP()      1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C  REG          REGION OF INTEREST IN A PARTICULAR DIVISION
C  SIGMA2()     1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 VARIANCE FOR EACH REGION
C  STR()        2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SZERO        STRESS TENSILE TEST POINT, So
C  VARY         CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO
C                 VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                 VARIATION;  3 - TRUNCATED NORMAL VARIATION
C  ZROREG       ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                 REGION

C  INITIALIZE COUNT AND NBND()

      COUNT = 0

      DO 10 L = 0, MAXREG
         NBND(L) = 0.0
   10 CONTINUE

C  INPUT DATA ON SPECIFIC MATERIAL FROM SPECFD AND ECHO TO SPECFO

      READ(1,*) DESCRP(0), FTY, FTU, NDIV, NPTS(0)
```

```
      IF (NPTS(0) .GT. MAXDAT) THEN
         WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &              'SPECIFIC MATERIAL'
         CALL TRMNAT
      ENDIF

      WRITE(3,900) DESCRP(0), FTY, FTU, NPTS(0)
      IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)

      WRITE(3,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

C     STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ

      FTUZ = FTU
      FTYZ = FTY

C     INPUT STRESS/LIFE INFORMATION — INCLUDING STRESS RATIO AND REGION
C     INFORMATION FROM SPECFD AND ECHO TO SPECFO

      DO 100 M = 1, NDIV

         READ (1,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE FOR RATIO:  ', RATIO
            CALL TRMNAT
         ENDIF

         IF (REG .GT. MAXREG) THEN
            WRITE(8,*) 'ERROR:  OVER REGION LIMIT IN SPECIFIC DATA SET'
            CALL TRMNAT
         ENDIF

         DO 110 I = (COUNT + 1), (COUNT + NUM)
            READ(1,*) RAWSTR(I,0), RAWNF(I,0)
  110    CONTINUE

C        CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT

         IF (RATIO .EQ. -1.0) THEN

C           STRESS RATIO IS CORRECT

            DO 120 I = (COUNT + 1), (COUNT + NUM)
               RATSTR(I,0) = RAWSTR(I,0)
  120       CONTINUE

         ELSE

C           STRESS RATIO TRANSFORMATION MUST BE DONE

            CALL CONVRT (0, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR,
     &                   RATIO, FTU, FTY)

         ENDIF

C        ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL

         DO 130 I = (COUNT + 1), (COUNT + NUM)

            WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
     &                   RATSTR(I,0), RAWNF(I,0)

            IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
     &                   RATIO, REG, RATSTR(I,0), RAWNF(I,0)

  130    CONTINUE

C        BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C        EXPCTD, AND PAREST

         K = NP(0,REG)

         DO 140 I = (COUNT + 1), (COUNT + NUM)
```

```
            K = K + 1
            LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
            LNNF(K,0,REG) = ALOG(RAWNF(I,0))
            STR(K,REG) = RATSTR(I,0)
            NF(K,REG) = RAWNF(I,0)
  140     CONTINUE

          IF (K .GT. MAXDAT) THEN
             WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &                  'SPECIFIC MATERIAL'
             CALL TRMNAT
          ENDIF

          NP(0,REG) = K
          REFNP(REG) = K
          COUNT = COUNT + NUM
  100 CONTINUE

      IF (NPTS(0) .NE. COUNT) THEN
         WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &              'INCORRECTLY SPECIFIED'
         WRITE(8,*) 'IN SPECIFIC DATA SET'
         CALL TRMNAT
      ENDIF

      READ(1,*) SZERO
      IF (NINT (SZERO) .GT. 0) THEN
         ZROREG = 0
      ELSE
         ZROREG = 1
      ENDIF
      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'SZERO = ', SZERO, ' ZROREG = ', ZROREG

C     INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION

      READ(1,*) NUMREG, NNODAT

      IF ((NUMREG + NNODAT) .GT. MAXREG)  THEN
         WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON NUMBER OF REGIONS'
         CALL TRMNAT
      ENDIF

      DO 150 L = ZROREG, (NUMREG + NNODAT)
         READ(1,*) NBND(L)
  150 CONTINUE

      READ(1,*) CZERO

      DO 160 L = 1, (NUMREG + NNODAT)
         READ(1,*) MPNT(L), MZERO(1,L), MZERO(2,L)
  160 CONTINUE

      WRITE(3,913)
      IF (ZROREG .EQ. 0) WRITE(3,914) SZERO
      IF (IOUT .EQ. 10) THEN
         WRITE(8,913)
         IF (ZROREG .EQ. 0) WRITE(8,914) SZERO
      ENDIF

      WRITE(3,915) NUMREG, NNODAT
      IF (IOUT .EQ. 10) WRITE(8,915) NUMREG, NNODAT

      DO 170 L = ZROREG, (NUMREG + NNODAT)
         WRITE(3,920) NBND(L)
         IF (IOUT .EQ. 10) WRITE(8,920) NBND(L)
  170 CONTINUE

      WRITE(3,925) CZERO
      IF (IOUT .EQ. 10) WRITE(8,925) CZERO

      DO 180 L = 1, (NUMREG + NNODAT)
```

```
         WRITE(3,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
         IF (IOUT .EQ. 10)
     &      WRITE(8,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
         IF ((VARY .EQ. 3) .AND. (MPNT(L) .EQ. 0)) THEN
            WRITE(8,*) 'ERROR:  NORMAL VARIATION REQUIRES A PRIOR ',
     &                 'RANGE ON M'
            CALL TRMNAT
         ENDIF
  180 CONTINUE

      IF (VARY .EQ. 3) THEN
C        READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
         WRITE(3,945)
         IF (IOUT .EQ. 10) WRITE(8,945)
         DO 190 L = 1, (NUMREG + NNODAT)
            READ(1,*) DELTA(L), MO(L), SIGMA2(L)
            WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
            IF (IOUT .EQ. 10)
     &         WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
            IF ((DELTA(L) .LT. 0.0) .OR.
     &         ((DELTA(L) .GT. 0.0) .AND. (MO(L) .LE. 0.0))) THEN
               WRITE(8,*) 'ERROR:  BAD VALUE FOR DELTA OR VALUE OF MO ',
     &                    'INCONSISTENT WITH DELTA IN REGION ', L
               CALL TRMNAT
            ENDIF
  190    CONTINUE
      ENDIF

      IF (MPROC .EQ. 1) THEN
         READ(1,*) KRATIO, LAMN
         WRITE(3,955) KRATIO, LAMN
         IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
      ENDIF


C  BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATD
C  AND THEN ECHO TO RELATO

      READ(5,*) NSETS

      IF (NSETS .GT. MAXSET) THEN
         WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF RELATED DATA SETS'
         CALL TRMNAT
      ENDIF

      WRITE(6,935) NSETS

      DO 200 J = 1, NSETS

         COUNT = 0

         IF (IOUT. EQ. 10) WRITE(8,*)'J =', J, ' NSETS =', NSETS

         READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)

         IF (NPTS(J) .GT. MAXDAT) THEN
            WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF POINTS IN ',
     &                 'SET ', J
            CALL TRMNAT
         ENDIF

         WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
         IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)

         WRITE(6,905)
         IF (IOUT .EQ. 10) WRITE(8,905)

         DO 300 M = 1, NDIV

            READ(5,*) NUM, RATIO, REG

            IF (ABS(RATIO) .GT. 1.0) THEN
               WRITE(8,*) 'ERROR:  INVALID VALUE OF RATIO:  ', RATIO
               CALL TRMNAT
            ENDIF
```

```
            IF (REG .GT. MAXREG) THEN
               WRITE(8,*)
     &            'ERROR:  OVER REGION LIMIT IN RELATED MATERIAL ', J
               CALL TRMNAT
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
               WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
            ENDIF

            DO 310 I = (COUNT + 1), (COUNT + NUM)
               READ(5,*) RAWSTR(I,J), RAWNF(I,J)
  310       CONTINUE

C  CHECK IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT

            IF (RATIO .EQ. -1.0) THEN

C              STRESS RATIO IS CORRECT

               DO 320 I = (COUNT + 1), (COUNT + NUM)
                  RATSTR(I,J) = RAWSTR(I,J)
  320          CONTINUE

            ELSE

C              STRESS RATIO TRANSFORMATION MUST BE DONE
               CALL CONVRT(J, (COUNT + 1), (COUNT + NUM), RAWSTR,
     &                     RATSTR, RATIO, FTU, FTY)

            ENDIF

C           RECORD BOTH S/N DATA SETS TO RELATO

            DO 330 I = (COUNT + 1), (COUNT + NUM)

               WRITE(6,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG,
     &                      RATSTR(I,J), RAWNF(I,J)

               IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,J), RAWNF(I,J),
     &                      RATIO, REG, RATSTR(I,J), RAWNF(I,J)

  330       CONTINUE

            K = NP(J,REG)

            DO 340 I = (COUNT + 1), (COUNT + NUM)

               K = K + 1
               LNSTR(K,J,REG) = ALOG(RATSTR(I,J))
               LNNF(K,J,REG) = ALOG(RAWNF(I,J))

  340       CONTINUE

            IF (K .GT. MAXDAT) THEN
               WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF POINTS ',
     &                    'IN SET ', J
               CALL TRMNAT
            ENDIF

            NP(J,REG) = K
            COUNT = COUNT + NUM

  300    CONTINUE

         IF (NPTS(J) .NE. COUNT) THEN
            WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &                 'INCORRECTLY SPECIFIED IN SET ', J
            CALL TRMNAT
         ENDIF

  200 CONTINUE
```

```
C  FORMAT STATEMENTS USED TO WRITE TO SPECFO AND RELATO

  900 FORMAT(////,13X,'MATERIAL INPUT',///,2X,'DESCRIPTION:',2X,A40,//,
     &         2X,'YIELD STRENGTH',18X,E11.5,//,2X,'ULTIMATE STRENGTH',
     &         15X,E11.5,//,2X,'NUMBER OF POINTS',16X,I2)

  905 FORMAT(//,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
     &         /,5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
     &         'STRESS',7X,'LIFE'/)

  910 FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)

  913 FORMAT(//)

  914 FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN So OF',
     &         5X,E11.5)

  915 FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA  ',
     &         /,2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
     &         /,2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
     &         '(CYCLES): ',/)

  920 FORMAT(10X,E9.3)

  925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
     &         'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
     &         //,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
     &         //,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',
     &         5X,'UPPER BOUND',/)

  930 FORMAT(6X,I1,11X,I1,12X,F7.4,9X,F7.4)

  935 FORMAT(20X,'NUMBER OF DATA SETS:',2X,I2,//,17X,
     &         'NOTE:  ALL Kt ASSUMED TO BE 1.0',////,23X,
     &         'TRANSFORMED DATA')

  940 FORMAT(///,2X,'DESCRIPTION:',2X,A40,
     &         //,2X,'YIELD STRENGTH',18X,F7.0,
     &         //,2X,'ULTIMATE STRENGTH',15X,F7.0,
     &         //,2X,'NUMBER OF POINTS',16X,I2)

  945 FORMAT(/,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS:',
     &         //,2X,'REGION',5X,'DELTA',8X,'mo',10X,'SIGMA2',/)

  950 FORMAT(5X,I1,5X,F7.2,5X,F7.4,5X,E11.5)

  955 FORMAT(//,2X,'MATERIALS PROCESS VARIATION INFORMATION',
     &         //,2X,'MEDK*/MEDK:',5X,E11.5,/,5X,'LAMBDAN:',5X,E11.5)

      RETURN
      END

C***********************************************************************


C  THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C  STRESS RATIO, R, IS NOT -1.0
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87       COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)

C  INPUTS:    J, NUM1, NUM2, STR, R, FTU, FTY
C  OUTPUTS:   RSTR

C     IMPLICIT NONE
```

```
      INTEGER MAXDAT, MAXSET

      PARAMETER (MAXDAT = 50, MAXSET = 5)

      COMMON IOUT

      INTEGER I, IOUT, J, NUM1, NUM2

      REAL     FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET),
     &         STR(MAXDAT, 0:MAXSET), TEST

C                   LIST OF VARIABLES
C
C  FTU          ULTIMATE STRENGTH OF MATERIAL (PSI)
C  FTY          YIELD STRENGTH OF MATERIAL (PSI)
C  I            CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C  IOUT         OUTPUT DUMP CONTROLLER
C  J            DATA SET OF INTEREST
C  MAXDAT       MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXSET       MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  NUM1         FIRST INDEX TO BE TRANSFORMED
C  NUM2         LAST INDEX TO BE TRANSFORMED
C  R            STRESS RATIO (R = -1.0 IS DESIRED)
C  RSTR()       STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
C  STR()        ARRAY CONTAINING STRESS VALUES (PSI) FOR S/N CURVE
C  TEST         Kt * Smax * (1 - R)/2 , TO BE COMPARED WITH FTY

C  Kt IS ASSUMED TO BE ONE

      DO 100 I = NUM1, NUM2

         TEST = STR(I,J) * (1.0 - R)/2.0

         IF (IOUT.EQ.10) WRITE(8,*) 'I =',I,' J =',J,' TEST =',TEST

         IF (TEST .GE. FTY) THEN

            RSTR(I,J) = TEST

            IF (IOUT.EQ.10) WRITE(8,*)'1:RSTR() =',RSTR(I,J)

         ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN

            RSTR(I,J) = TEST/(1.0 -((FTY - TEST)/FTU))

            IF (IOUT.EQ.10) WRITE(8,*)'2:RSTR() =',RSTR(I,J)

         ELSE

            RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
     &                  /(2.0 * FTU)))

            IF (IOUT.EQ.10) WRITE(8,*)'3:RSTR() =',RSTR(I,J)

         END IF

  100 CONTINUE

      RETURN
      END


C**********************************************************************


C  SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C  AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C  X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87     COMMENTS:  13JUL89
```

```
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY,
     &                   SY2, DD, SWHAT2, SUHAT2, NPPR)

C  INPUTS:   NUMREG, NSETS, NP, LNSTR, LNNF
C  OUTPUTS:  SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

      PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER IOUT, J, K, L, NP(0:MAXSET, MAXREG), NPPR(MAXREG),
     &        NSETS, NUMREG

      REAL    BB(MAXREG), DD(MAXREG), DIFFX(MAXDAT, 0:MAXSET),
     &        DIFFY(MAXDAT, 0:MAXSET), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MEANX(0:MAXSET),
     &        MEANY(0:MAXSET), SUHAT2(MAXREG), SWHAT2(MAXREG),
     &        SX2(MAXREG), SXY(MAXREG), SY2(MAXREG)


C                    LIST OF VARIABLES
C
C  BB()      1-D ARRAY CONTAINING SXY(L)/SY2(L) FOR EACH REGION
C  DD()      1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DIFFX()   2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L)
C               AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C  DIFFY()   2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L)
C               AND MEANY(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C  IOUT      OUTPUT DUMP CONTROLLER
C  J         CONTROLS DO LOOP FOR EACH DATA SET
C  K         CONTROLS DO LOOP FOR EACH POINT IN A REGION
C  L         CONTROLS DO LOOP FOR EACH REGION
C  LNNF()    3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
C  LNSTR()   3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C  MAXDAT    MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET    MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MEANX()   1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION
C              L AND DATA SET J (X = Ln S)
C  MEANY()   1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
C              L AND DATA SET J (Y = Ln N)
C  NP()      2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C              SET IN EACH REGION
C  NPPR()    1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C              ALL DATA SETS IN A REGION (Number of Points Per Region)
C  NSETS     NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  SUHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C              REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C  SWHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C              REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C  SX2()     1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C              (X = Ln S)
C  SXY()     1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
C              EACH REGION (X = Ln S, Y = Ln N)
C  SY2()     1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C              (Y = Ln N)

C  INITIALIZE ARRAYS

      DO 50 L = 1, MAXREG
        SY2(L) = 0.0
        SX2(L) = 0.0
        SXY(L) = 0.0
        SWHAT2(L) = 0.0
        SUHAT2(L) = 0.0
```

```
            BB(L) = 0.0
            DD(L) = 0.0
            NPPR(L) = 0
   50 CONTINUE

         DO 60 J = 0, MAXSET
            DO 70 K = 1, MAXDAT
               DIFFY(K,J) = 0.0
               DIFFX(K,J) = 0.0
   70       CONTINUE
            MEANY(J) = 0.0
            MEANX(J) = 0.0
   60 CONTINUE

C  NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION

         DO 100 L = 1, NUMREG

            DO 200 J = 0, NSETS
C              FIRST CALCULATE SAMPLE X AND Y MEANS
C              FOR DATA SET J IN REGION L
               MEANY(J) = 0.0
               MEANX(J) = 0.0
               IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' J =', J,
     &                        ' NP =', NP(J,L)

               DO 250 K = 1, NP(J,L)
                  MEANY(J) = MEANY(J) + LNNF(K,J,L)
                  MEANX(J) = MEANX(J) + LNSTR(K,J,L)
                  IF (IOUT .EQ. 10) WRITE(8,*) 'LNNF =', LNNF(K,J,L),
     &                ' LNSTR =', LNSTR(K,J,L)
  250          CONTINUE

               MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
               MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
               IF (IOUT .EQ. 10) WRITE(8,*) 'MEANY(J) =', MEANY(J),
     &            ' MEANX(J) =', MEANX(J)

C              NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY,
C              OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
C              DATA SET IN REGION L

               DO 300 K = 1, NP(J,L)
                  DIFFY(K,J) = LNNF(K,J,L) - MEANY(J)
                  DIFFX(K,J) = LNSTR(K,J,L) - MEANX(J)
                  SY2(L) = SY2(L) + DIFFY(K,J) ** 2
                  SX2(L) = SX2(L) + DIFFX(K,J) ** 2
                  SXY(L) = SXY(L) + DIFFX(K,J) * DIFFY(K,J)
                  IF (IOUT .EQ. 10) THEN
                     WRITE(8,*) 'K =', K, ' DIFFY(K,J) =', DIFFY(K,J),
     &                          ' DIFFX(K,J) =', DIFFX(K,J)
                     WRITE(8,*) 'SY2(L) =', SY2(L), ' SX2(L) =', SX2(L),
     &                          ' SXY(L) =', SXY(L)
                  ENDIF
  300          CONTINUE

               NPPR(L) = NPPR(L) + NP(J,L) - 1
               IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) =', NPPR(L)
  200       CONTINUE

            IF (SXY(L) .GE. 0.0) THEN
C              LIFE WILL INCREASE WITH INCREASING STRESS — INVALID FOR
C              OUR MODEL
               WRITE(8,*) 'ERROR: SXY >= 0 IN REGION', L
               CALL TRMNAT
            ENDIF

            NPPR(L) = NPPR(L) - 1

            IF (NPPR(L) .LE. 0) THEN
               WRITE(8,*) 'ERROR:  TOO FEW POINTS FOR REGRESSION IN ',
     &                    'REGION ',L
               CALL TRMNAT
            ENDIF
```

```
         SY2(L) = SY2(L) / FLOAT(NPPR(L))
         SX2(L) = SX2(L) / FLOAT(NPPR(L))
         SXY(L) = SXY(L) / FLOAT(NPPR(L))

C        NOW CALCULATE THE RESIDUAL VARIANCES, SWHAT2, SUHAT2, FOR EACH
C        REGION FROM THE Y ON X AND X ON Y REGRESSIONS

         DD(L) = SXY(L) / SX2(L)
         BB(L) = SXY(L) / SY2(L)
         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
     &                 ' SX2(L) =', SX2(L)
            WRITE(8,*) 'SXY(L) =', SXY(L), ' DD(L) =', DD(L),
     &                 ' BB(L) =', BB(L)
         ENDIF

         DO 400 J = 0, NSETS
            IF (IOUT .EQ. 10) WRITE(8,*) 'J =', J, ' NP(J,L) =', NP(J,L)

            DO 500 K = 1, NP(J,L)
               SWHAT2(L) = SWHAT2(L)
     &                     + (DIFFY(K,J) - DD(L) * DIFFX(K,J)) ** 2
               SUHAT2(L) = SUHAT2(L)
     &                     + (DIFFX(K,J) - BB(L) * DIFFY(K,J)) ** 2
               IF (IOUT .EQ. 10) WRITE(8,*) 'K =', K, ' SWHAT2(L) =',
     &            SWHAT2(L), ' SUHAT2(L) =', SUHAT2(L)
 500        CONTINUE

 400     CONTINUE

         SWHAT2(L) = SWHAT2(L) / FLOAT(NPPR(L))
         SUHAT2(L) = SUHAT2(L) / FLOAT(NPPR(L))
         IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) =', NPPR(L),
     &      ' SWHAT2(L) =', SWHAT2(L), ' SUHAT2(L) =', SUHAT2(L)
 100  CONTINUE

      RETURN
      END


C**************************************************************************


C   SUBROUTINE INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, Io, ON
C   C; AND THE 95% CONFIDENCE INTERVAL, Jo, ON M
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  5OCT87    COMMENTS:  15SEP89
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                   JZERO, MCHAT)

C   INPUTS:    NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C   OUTPUTS:   IZERO, JZERO, MCHAT
C   SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER CHITAB, MAXREG, TTAB

      PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)

      COMMON  IOUT

      INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG

      REAL     ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
     &         IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
     &         SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,
     &         SX2(MAXREG), T, T025(TTAB)
```

```
      DATA (CHI025(I), I = 1, 75) /
     &             0.000982069, 0.506356, 0.215795, 0.484419, 0.831211,
     &             1.237347, 1.68987, 2.17973, 2.70039, 3.24697,
     &             3.81575,  4.40379, 5.00874, 5.62872, 6.26214,
     &             6.90766,  7.56418, 8.23075, 8.90655, 9.59083,
     &             10.28293, 10.9823, 11.6885, 12.4011, 13.1197,
     &             13.8439,  14.5733, 15.3079, 16.0471, 16.7908,
     &             17.53,    18.28,   19.04,   19.80,   20.56,
     &             21.33,    22.10,   22.87,   23.65,   24.4331,
     &             25.21,    25.99,   26.78,   27.57,   28.36,
     &             29.15,    29.95,   30.75,   31.55,   32.3574,
     &             33.16,    33.96,   34.77,   35.58,   36.39,
     &             37.21,    38.02,   38.84,   39.66,   40.4817,
     &             41.30,    42.12,   42.95,   43.77,   44.60,
     &             45.43,    46.26,   47.09,   47.92,   48.7576,
     &             49.59,    50.42,   51.26,   52.10,   52.94 /
      DATA (CHI025(I), I = 76, 150) /
     &             53.78,    54.62,   55.46,   56.30,   57.1532,
     &             57.80,    58.84,   59.69,   60.54,   61.39,
     &             62.24,    63.09,   63.94,   64.79,   65.6466,
     &             66.50,    67.35,   68.21,   69.07,   69.92,
     &             70.78,    71.64,   72.50,   73.36,   74.2219,
     &             75.08,    75.94,   76.80,   77.67,   78.53,
     &             79.40,    80.27,   81.13,   82.00,   82.87,
     &             83.73,    84.60,   85.47,   86.34,   87.21,
     &             88.08,    88.95,   89.83,   90.70,   91.57,
     &             92.45,    93.32,   94.19,   95.07,   95.94,
     &             96.82,    97.70,   98.57,   99.45,   100.33,
     &             101.21,   102.09,  102.97,  103.85,  104.73,
     &             105.61,   106.49,  107.37,  108.25,  109.14,
     &             110.02,   110.90,  111.79,  112.67,  113.56,
     &             114.44,   115.33,  116.21,  117.10,  117.98 /

      DATA (CHI975(I), I = 1, 75) /
     &             5.02389,  7.37776, 9.34840, 11.1433, 12.8325,
     &             14.4494,  16.0128, 17.5346, 19.0228, 20.4831,
     &             21.9200,  23.3367, 24.7356, 26.1190, 27.4884,
     &             28.8454,  30.1910, 31.5264, 32.8523, 34.1696,
     &             35.4789,  36.7807, 38.0757, 39.3641, 40.6465,
     &             41.9232,  43.1944, 44.4607, 45.7222, 46.9792,
     &             48.23,    49.48,   50.72,   51.96,   53.20,
     &             54.44,    55.67,   56.89,   58.12,   59.3417,
     &             60.56,    61.77,   62.99,   64.20,   65.41,
     &             66.62,    67.82,   69.02,   70.22,   71.4202,
     &             72.61,    73.81,   75.00,   76.19,   77.38,
     &             78.57,    79.75,   80.93,   82.12,   83.2976,
     &             84.48,    85.65,   86.83,   88.00,   89.18,
     &             90.35,    91.52,   92.69,   93.86,   95.0231,
     &             96.19,    97.35,   98.52,   99.68,   100.84 /
      DATA (CHI975(I), I = 76, 150) /
     &             102.00,   103.16,  104.31,  105.47,  106.629,
     &             107.78,   108.94,  110.09,  111.24,  112.39,
     &             113.54,   114.69,  115.84,  116.99,  118.136,
     &             119.28,   120.43,  121.57,  122.72,  123.86,
     &             125.00,   126.14,  127.28,  128.42,  129.561,
     &             130.70,   131.84,  132.98,  134.11,  135.25,
     &             136.38,   137.52,  138.65,  139.79,  140.92,
     &             142.05,   143.18,  144.31,  145.44,  146.57,
     &             147.70,   148.83,  149.96,  151.09,  152.21,
     &             153.34,   154.47,  155.59,  156.72,  157.84,
     &             158.97,   160.09,  161.21,  162.33,  163.46,
     &             164.58,   165.70,  166.82,  167.94,  169.06,
     &             170.18,   171.30,  172.41,  173.53,  174.65,
     &             175.77,   176.88,  178.00,  179.12,  180.23,
     &             181.35,   182.46,  183.58,  184.69,  185.80 /

C  VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:
C
C     1 - 30, 40, 50, 60, 70, 80, 90, 100 -- Theil, pp. 718-719
C
C     31-39, 41-49, 51-59, 61-69, 71-79, 81-89, 91-99, 101-150
C        -- CALCULATED USING CUBE RULE APPROXIMATION
```

```
      DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447,
     &             2.365, 2.306, 2.262, 2.228, 2.201, 2.179,
     &             2.160, 2.145, 2.131, 2.120, 2.110, 2.101,
     &             2.093, 2.086, 2.080, 2.074, 2.069, 2.064,
     &             2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /

C                         LIST OF VARIABLES
C
C   ARG         INTERMEDIATE CALCULATION VARIABLE
C   CHI025()    TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C   CHI975()    TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C   CHITAB      MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
C   DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C   I           CONTROLS LOOP FOR CHI025() AND CHI975()
C   IOUT        OUTPUT DUMP CONTROLLER
C   IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                 FOR EACH REGION
C   JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C   L           CONTROLS DO LOOP FOR EACH REGION
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
C                 MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL
C                 DATA SETS IN A REGION (Number of Points Per Region)
C   NUM         EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   SUHAT       EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C   SUHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SWHAT       EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C   SWHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SX          EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
C   SX2()       1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                 (X = Ln S)
C   T           VALUE OF T025() USED IN CALCULATIONS
C   T025()      TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
C   TTAB        MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C   INITIALIZE IZERO, JZERO AND MCHAT

      DO 50 L = 1, MAXREG
         IZERO(1,L) = 0.0
         IZERO(2,L) = 0.0
         JZERO(1,L) = 0.0
         JZERO(2,L) = 0.0
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
   50 CONTINUE

C   CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

      DO 75 L = 1, NUMREG
         IF (NPPR(L) .GT. CHITAB) THEN
            WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON DEGREES OF FREEDOM ',
     &                 'IN CHI-SQUARE TABLE, IN REGION ', L
            CALL TRMNAT
         ENDIF
   75 CONTINUE

C   ASSIGN VALUES TO NUM, T, SWHAT, SUHAT AND THEN CALCULATE
C   CONFIDENCE INTERVALS FOR EACH REGION

      DO 100 L = 1, NUMREG

         NUM = NPPR(L)

         IF (NUM .LT. 31) THEN
```

```
            T = T025(NUM)
         ELSE
            T = T025(NUM)
         ENDIF

         SWHAT = SWHAT2(L) ** 0.5
         SUHAT = SUHAT2(L) ** 0.5
         SX = (NUM * SX2(L)) ** 0.5

C        CALCULATE ESTIMATED VALUES OF M AND C

         ARG = T * SWHAT / SX
         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SUHAT

C        CALCULATE CONFIDENCE INTERVALS

         IZERO(1,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI975(NUM)) ** 0.5
         IZERO(2,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI025(NUM)) ** 0.5
         JZERO(1,L) = MCHAT(1,L) - ARG
         JZERO(2,L) = MCHAT(1,L) + ARG


         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L =', L, ' NPPR =', NPPR(L), ' NUM =', NUM
            WRITE(8,*) 'SWHAT2 =', SWHAT2(L), ' SWHAT =', SWHAT
            WRITE(8,*) 'SUHAT2 =', SUHAT2(L), ' SUHAT =', SUHAT
            WRITE(8,*) 'SX2 =', SX2(L), ' SX =', SX
            WRITE(8,*) 'CHI025 =', CHI025(NUM), ' CHI975 =', CHI975(NUM)
            WRITE(8,*) 'T =', T, ' DD =', DD(L), ' ARG =', ARG
            WRITE(8,*) 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =',
     &                  IZERO(2,L)
            WRITE(8,*) 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =',
     &                  JZERO(2,L)
            WRITE(8,*) 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =',
     &                  MCHAT(2,L)
         ENDIF
  100 CONTINUE

      RETURN
      END


C*************************************************************************


C  SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C  THE Co GIVEN BY THE USER
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  8OCT87     COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

C  INPUTS:    NUMREG, CZERO, SX2, SXY, SY2
C  OUTPUTS:   MCPNT, MC

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, MCPNT(MAXREG), NUMREG

      REAL     ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
     &         SXY(MAXREG), SY2(MAXREG)
```

```
C                    LIST OF VARIABLES
C
C   ARG1       INTERMEDIATE CALCULATION VARIABLE
C   ARG2       INTERMEDIATE CALCULATION VARIABLE
C   CZERO      EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C   CZERO2     EQUAL TO CZERO ** 2
C   IOUT       OUTPUT DUMP CONTROLLER
C   L          CONTROLS DO LOOP FOR EACH REGION
C   MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C   MC()       2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
C                CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA - MC(1,L) IS
C                THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
C   MCPNT()    1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MC() FOR EACH REGION
C   NUMREG     NUMBER OF REGIONS OF INTEREST
C   SX2()      1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                (X = Ln S)
C   SXY()      1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR
C                EACH REGION (X = Ln S, Y = Ln N)
C   SY2()      1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                (Y = Ln N)


C     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         MCPNT(L) = 0
         MC(1,L) = 0.0
         MC(2,L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATIONS

      CZERO2 = CZERO ** 2

      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2

      DO 100 L = 1, NUMREG

         ARG1 = SX2(L) - CZERO2
         ARG2 = 0.0

         IF (CZERO .EQ. 0.0) THEN

C           THEN NO M CONSTRAINT IS REQUIRED

            MCPNT(L) = 0

         ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN

C           THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M

            MCPNT(L) = 1
            MC(1,L) = - SY2(L) / (2.0 * SXY(L))

         ELSE

C           THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
C           COMMON CALCULATIONS

            ARG2 = (SXY(L) ** 2 - SY2(L) * ARG1)
            IF (ARG2 .LT. 0.0) THEN
C              ARG2 IS NEGATIVE - IMPLIES M IS COMPLEX
               WRITE(8,*) 'ERROR:  Co TOO LOW'
               CALL TRMNAT
            ELSE
               ARG2 = ARG2 ** 0.5
            ENDIF

            IF (SX2(L) .LT. CZERO2) THEN
```

```
C               AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M

                MCPNT(L) = 1
                MC(1,L) = (- SXY(L) - ARG2) / ARG1

             ELSE

C               SX2(L) .GT. CZERO2 — THIS TIME THE M CONSTRAINT IS A RANGE

                MCPNT(L) = 2
                MC(1,L) = (- SXY(L) - ARG2) / ARG1
                MC(2,L) = (- SXY(L) + ARG2) / ARG1

             ENDIF

          ENDIF

  100 CONTINUE

      IF (IOUT .EQ. 10) THEN

         DO 200 L = 1, NUMREG

            WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
            WRITE(8,*) 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)

  200    CONTINUE

      ENDIF

      RETURN
      END

C************************************************************************

C  SUBROUTINE GTPVAR CALCULATES THE EXTENT OF DEPARTURE FROM THE MULTIPLE
C  HEAT MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88     COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)

C  INPUTS:   NSETS, NP, NUMREG, LAMN, MCHAT
C  OUTPUTS:  PVAR

C     IMPLICIT NONE

      INTEGER MAXREG, MAXSET

      PARAMETER (MAXREG = 3, MAXSET = 5)

      COMMON  IOUT

      INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG),
     &        NUMREG, TOTAL

      REAL    LAMN, MCHAT(2, MAXREG), PSIG2(MAXREG), PVAR, SUM


C                    LIST OF VARIABLES
C
C  IOUT          OUTPUT DUMP CONTROLLER
C  J             CONTROLS DO LOOP FOR EACH DATA SET
C  L             CONTROLS DO LOOP FOR EACH REGION
C  LAMN          LAMBDA-N — RATIO OF Var (Ln N given S) / (m**2 C**2),
```

```
C                 CONSTANT OVER REGIONS AND COMPONENTS
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET       MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                 MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  NP()         2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                 SET IN EACH REGION
C  NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUM()        EQUAL TO Nj-1 FOR EACH REGION WHERE Nj IS THE SUM OF THE
C                 NUMBER OF POINTS IN EACH DATA SET
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  PSIG2()      1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
C                 VARIATION IN EACH REGION
C  PVAR         THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C                 CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  SUM          WEIGHTED SUM OF THE PSIG2s - USED TO CALCULATE A WEIGHTED
C                 AVERAGE
C  TOTAL        SUM OF NUM() OVER ALL REGIONS

C     INITIALIZE VARIABLES

      SUM = 0.0
      TOTAL = 0.0

      DO 50 L = 1, MAXREG
         PSIG2(L) = 0.0
         NUM(L) = 0
   50 CONTINUE

      DO 100 L = 1, NUMREG
         DO 150 J = 0, NSETS
            NUM(L) = NUM(L) + NP(J,L)
  150    CONTINUE
         NUM(L) = NUM(L) - 1
         TOTAL = TOTAL + NUM(L)
  100 CONTINUE

      DO 200 L = 1, NUMREG
         PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
         SUM = SUM + PSIG2(L) * NUM(L)
  200 CONTINUE

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'LAMN = ', LAMN
         DO 300 L = 1, NUMREG
            WRITE(8,*) 'L = ', L, ' NUM = ', NUM(L)
            WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
  300    CONTINUE
         WRITE(8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
      ENDIF

      PVAR = SUM / FLOAT (TOTAL)

      RETURN
      END

C***********************************************************************


C  SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
C  M AND Co WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C  TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:   2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
     &                   MCHAT, RANGEM)
```

```
C  INPUTS:   NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C  OUTPUTS:  RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
     &        MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

C                          LIST OF VARIABLES
C
C  IOUT       OUTPUT DUMP CONTROLLER
C  JZERO()    2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C               FOR EACH REGION
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LOWER      LOWER BOUND OF INTERSECTION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()       2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C               REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C               - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C               BOUND
C  MCHAT()    2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C               FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
C               FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()    1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MC() FOR EACH REGION
C  MPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MZERO() FOR EACH REGION
C  MZERO()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C               EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C               IS THE UPPER BOUND
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  RANGEM()   2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C               FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C               RANGEM(2,L) IS THE UPPER BOUND
C  UPPER      UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         RANGEM(1,L) = 0.0
         RANGEM(2,L) = 0.0
   50 CONTINUE

C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
         ENDIF

         IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS NO EXOGENOUS INFORMATION
C           ASSUME RANGE TO BE Jo

            RANGEM(1,L) = JZERO(1,L)
            RANGEM(2,L) = JZERO(2,L)

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' JZERO(1,L) = ', JZERO(1,L)
```

```
            WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN

C        NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE
C        TO Co, ADJUST THE LOWER BOUND OF Jo ACCORDINGLY

         LOWER = AMAX1(JZERO(1,L), MC(1,L))
         UPPER = JZERO(2,L)
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN

C        THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE
C        CORRESPONDING TO THE Co CONSTRAINT, ADJUST Jo ACCORDINGLY

         LOWER = AMAX1(JZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF (MPNT(L) .EQ. 1) THEN

C        THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER
C        INFORMATION:  ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR

         RANGEM(1,L) = MZERO(1,L)
         RANGEM(2,L) = 0.0

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN

C        THERE IS A PRIOR RANGE ON M, BUT NO Co CONSTRAINT
C        USE INTERSECTION BETWEEN Jo AND Mo

         LOWER = AMAX1(JZERO(1,L), MZERO(1,L))
         UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mo'
```

```
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN
C
C        THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C        CONSTRAINT, INTERSECT Jo AND Mo, ADJUSTING THE LOWER BOUND
C        BY Mc ACCORDINGLY

         LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                 'AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANEGM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN
C
C        THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO Co CONSTRAINT
C        INTERSECT THESE TWO RANGES WITH Jo

         LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                 'AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSE

         WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
         CALL TRMNAT
```

```
         ENDIF

C        RESTRICT RANGE TO BE NON-NEGATIVE

         RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

         IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &       .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END

C***********************************************************************

C  SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C  WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  INPUTS:   RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C  OUTPUTS:  RANGEM, MCHAT, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

      REAL    MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)


C                      LIST OF VARIABLES
C
C  IOUT          OUTPUT DUMP CONTROLLER
C  L             CONTROLS DO LOOP FOR EACH REGION
C  LL            EQUAL TO NUMREG FOR A SET OF CALCULATIONS
C  MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C  MCHAT()       2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C                  C FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
C                  MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND
C                  MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MPNT()        1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                  MZERO() FOR EACH REGION
C  MZERO()       2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                  EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
```

```
C                 IS UPPER BOUND
C  NNODAT       Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND

      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
            ENDIF

C           SPECIFY E(M) OF POSTERIOR FOR SAKE OF
C           CALCULATIONS IN SUBROUTINE EXPCTD

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE

      RETURN
      END


C***************************************************************************


C  SUBROUTINE CONCAV ADJUSTS THE UPPER BOUNDS OF THE POSTERIOR CREDIBILITY
C  RANGES ON M TO BE CONSISTENT WITH CONCAVITY CONSTRAINTS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  2FEB88       FORMAT/COMMENTS:  15SEP89
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONCAV (NUMREG, RANGEM)

C  INPUTS:   NUMREG, RANGEM
C  OUTPUTS:  RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG
```

```
      REAL     RANGEM(2, MAXREG), TESTM

C                               LIST OF VARIABLES
C
C   IOUT          OUTPUT DUMP CONTROLLER
C   L             CONTROLS DO LOOP FOR EACH REGION
C   MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C   NUMREG        NUMBER OF REGIONS OF INTEREST
C   RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                   FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                   RANGEM(2,L) IS THE UPPER BOUND
C   TESTM        UPPER BOUND OF RANGE ON M IN REGION L-1 - USED DURING
C                  CONCAVITY ADJUSTMENT

C     ADJUST RANGE TO INSURE CONCAVITY

      DO 100 L = NUMREG, 2, -1

         IF (RANGEM(2,L-1) .EQ. 0.0) THEN
C           RANGE IS A POINT IN REGION L-1
            IF (RANGEM(1,L-1) .GT. AMAX1(RANGEM(1,L),RANGEM(2,L))) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &           ' IS INCONSISTENT WITH POINT POSTERIOR IN REGION ',L-1
               CALL TRMNAT
            ENDIF
         ELSE
C           RANGE IS AN INTERVAL IN REGION L-1
            TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
            IF (TESTM .LT. RANGEM(1,L-1)) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &           ' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
     &           'REGION ', L-1
               CALL TRMNAT
            ELSE
               RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
            ENDIF
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,L-1) =', RANGEM(1,L-1),
     &                 ' RANGEM(2,L-1) =', RANGEM(2,L-1)
            WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                 ' RANGEM(2,L) =', RANGEM(2,L)
            WRITE(8,*) 'TESTM =', TESTM, ' L = ', L
         ENDIF

  100 CONTINUE

      RETURN
      END


C***************************************************************************


C   SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS
C   BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR Co
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  5OCT87      COMMENTS:  1DEC87
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C   INPUTS:    NUMREG, RANGEM
C   IOUTPUT:   MEDM

C     IMPLICIT NONE
```

```
      INTEGER   MAXREG
      PARAMETER (MAXREG = 3)
      COMMON    IOUT
      INTEGER   IOUT, L, NUMREG
      REAL      LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)

C                   LIST OF VARIABLES
C
C   IOUT         OUTPUT DUMP CONTROLLER
C   L            CONTROLS DO LOOP FOR EACH REGION
C   LOWERM       LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
C                  TO BE USED IN MEDIAN CALCULATION
C   MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C   MEDM()       1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C   NUMREG       NUMBER OF REGIONS OF INTEREST
C   RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                  FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                  RANGEM(2,L) IS THE UPPER BOUND

C     INITIALIZE ARRAY MEDM
      DO 50 L = 1, MAXREG
         MEDM(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATIONS FOR EACH REGION
      DO 100 L = 1, NUMREG
         IF (RANGEM(2,L) .EQ. 0.0) THEN
C           RANGE IS A POINT
            MEDM(L) = RANGEM(1,L)
         ELSEIF (L .EQ. 1) THEN
C           WE ARE IN REGION ONE - NOT AFFECTED BY OTHER REGIONS
C           - MEDIAN WILL JUST BE AVERAGE OF RANGEM VALUES
            MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
         ELSE
C           MUST TAKE MEDIAN OF REGION L-1 INTO ACCOUNT
            LOWERM = AMAX1(RANGEM(1,L), MEDM(L-1))
            MEDM(L) = (LOWERM + RANGEM(2,L)) / 2.0
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
            WRITE(8,*) 'LOWERM = ', LOWERM, ' MEDM(L) = ', MEDM(L)
         ENDIF
  100 CONTINUE

      RETURN
      END

C*************************************************************************
```

```
C   SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES OF THE S/N
C   CURVE PARAMETERS
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  13FEB89      FORMAT/COMMENTS:  15SEP89
C      VERSION:  MATCHR V8.3, V8.4, V8.5    MATGRM V4.3, V4.4, V4.5
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.

      SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
     &                   ZROREG, NBND, BIGK1, BZHAT)

C   INPUTS:  NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
C   OUTPUTS:  BIGK1, BZHAT
C   SUBPROGRAMS:  TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG

      REAL      BIGK(0:MAXREG), BIGK1, BZHAT, FACTR, KHAT, MEANZ,
     &          MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
     &          NF(MAXDAT, MAXREG), SBND(0:MAXREG), STR(MAXDAT, MAXREG),
     &          SZ2, SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)

C                      LIST OF VARIABLES
C
C   BIGK()      1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C                 EACH REGION
C   BIGK1       EQUAL TO BIGK(1)
C   BZHAT       E(BETAO)
C   FACTR       A SCALE FACTOR = PHI * KRATIO * Z
C   IOUT        OUTPUT DUMP CONTROLLER
C   KHAT        E(k)
C   L           CONTROLS DO LOOP FOR EACH REGION
C   MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MEANZ       SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C   MEDM()      1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C   MM()        1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C   NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST
C   NCOMPS      Number of COMPonents - 1 FOR STRESS AND STRAIN WHEN DECOMPOSED
C                 DATA UNAVAILABLE - 2 FOR DECOMPOSED STRAIN DATA
C   NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C                 DATA SET
C   NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR
C                 THE SPECIFIC MATERIAL S/N DATA SET
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   SBND()      1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C                 CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
C                 CONTAINED IN NBND()
C   STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C                 DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SZ2         SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C   SZERO       STRESS TENSILE TEST POINT, So
C   TRBIGK()    1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
C                 TRBIGK(i) = BIGK(i)
C   ZROREG      ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C   ZZ()        1-D ARRAY CONTAINING TRANSFORMED S-N DATA, Z = F(STR,NF,NBND,MM)
```

```
C  INITIALIZE VARIABLES

      DO 50 L = 0, MAXREG
         MM(L) = 0.0
   50 CONTINUE

C  CREATE MM() ARRAY FROM MEDM() ARRAY

      DO 100 L = 1, NUMREG
         MM(L) = MEDM(L)
  100 CONTINUE

C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

      CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C  CALCULATE BETAo AND k

      CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

C  CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

      CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

      BIGK1 = BIGK(1)

C  CALCULATE BOUNDARIES OF STRESS REGIONS

      CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C  CALCULATE Ko AND Mo FOR THE NO DATA REGION TO THE LEFT IF REQUIRED

      DO 150 L = ZROREG, NUMREG
         TRBIGK(L) = BIGK(L)
  150 CONTINUE

      IF (ZROREG .EQ. 0) THEN
         FACTR = 1.0
         CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK,
     &              FACTR, NUMREG)
      ENDIF

C  WRITE RESULTS TO FILE

      IF (NCOMPS .EQ. 1) THEN

         WRITE(7,900) NUMREG, BZHAT, KHAT
         IF (IOUT .EQ. 10) WRITE(8,900) NUMREG, BZHAT, KHAT

         DO 200 L = ZROREG, NUMREG
            WRITE(7,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
            IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
     &                                     NBND(L), SBND(L)
  200    CONTINUE

         WRITE(7,920)

      ELSE

         WRITE(7,930) MM(1), BIGK(1), KHAT

      ENDIF

C  FORMAT STATEMENTS

  900 FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',//,2X,
     &        'NUMBER OF REGIONS:',I4,5X,'E(BETAo) =',F8.4,5X,'E(k) =',
     &        F8.4,//,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',7X,
     &        'STRESS BOUND',/)
```

```
  910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E9.3,9X,E11.5)

  920 FORMAT(///)

  930 FORMAT(//,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
     &        //,11X,'m',14X,'K',13X,'E(k)',
     &        //,7X,F8.5,5X,E12.5,6X,F7.4,/)

      RETURN
      END

C***********************************************************************

C   SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C   MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  21JUN88     COMMENTS:  13JUL89
C      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA,
     &                  MO, SIGMA2, MCHAT, MU, SIG)

C   INPUTS:  NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2
C   OUTPUTS:  MCHAT, MU, SIG

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG, NPPR(MAXREG)

      REAL     ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG),
     &         MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG),
     &         SUHAT2(MAXREG), SUMX2, SWHAT2(MAXREG), SX2(MAXREG)

C              LIST OF VARIABLES
C
C   ARG        INTERMEDIATE CALCULATION VARIABLE
C   DD()       1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C   DELTA()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                SIG() CALCULATION
C   IOUT       OUTPUT DUMP CONTROLLER
C   L          CONTROLS DO LOOP FOR EACH REGION
C   MAXREG     MAXIMUM NUMBER OF REGION ALLOWED
C   MCHAT()    2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR
C                EACH REGION, BASED ON MATERIALS DATA ONLY - MCHAT(1,L) =
C                - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT,
C                THE ESTIMATE FOR C
C   MO()       1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C   MU()       1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C   NPPR()     1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL
C                DATA SETS IN A REGION (Number of Points Per Region)
C   NUMREG     NUMBER OF REGIONS OF INTEREST
C   SIG()      1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C   SIGMA2()   1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                VARIANCE FOR EACH REGION
C   SUHAT2()   1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C   SUMX2      EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
C   SWHAT2()   1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
```

```
C  SX2()        1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                  (X = Ln S)

C     INITALIZE ARRAYS

      DO 50 L = 1, MAXREG
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
         MU(L) = 0.0
         SIG(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATION FOR EACH REGION

      DO 100 L = 1 , NUMREG

         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SQRT (SUHAT2(L))
         SUMX2 = NPPR(L) * SX2(L)
         ARG = SUMX2 + DELTA(L)

         IF (DELTA(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
C           USE THE ESTIMATE OF M
            MU(L) = MCHAT(1,L)
         ELSE
C           UPDATE THE ESTIMATE OF M WITH Mo USING DELTA
            MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
         ENDIF

         IF (SIGMA2(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
C           USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
            SIG(L) = SQRT (SWHAT2(L) / ARG)
         ELSE
            SIG(L) = SQRT (SIGMA2(L) / ARG)
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ',
     &                 MCHAT(1,L)
            WRITE(8,*) 'SUHAT2 = ', SUHAT2(L),  ' MCHAT2 = ',
     &                 MCHAT(2,L)
            WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
     &                ' SUMX2 = ', SUMX2
            WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
            WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
            WRITE(8,*) 'SWHAT2 = ', SWHAT2(L), ' SIGMA2 = ', SIGMA2(L),
     &                ' SIG = ', SIG(L)
         ENDIF

  100 CONTINUE

      RETURN
      END

C***********************************************************************


C  SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND Co TO
C  OBTAIN POSTERIOR RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  10FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  INPUTS:   NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C  OUTPUTS:  RANGEM
C  SUBPROGRAMS:  TRMNAT
```

```
C     IMPLICIT NONE
      INTEGER MAXREG
      PARAMETER (MAXREG = 3)
      COMMON  IOUT
      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
      REAL    LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
     &        RANGEM(2, MAXREG), UPPER

C                         LIST OF VARIABLES
C
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LOWER      LOWER BOUND OF INTERSECTION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()       2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C               REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C               - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C               BOUND
C  MCHAT()    2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C               FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
C               FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()    1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MC() FOR EACH REGION
C  MPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MZERO() FOR EACH REGION
C  MZERO()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C               EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C               IS THE UPPER BOUND
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  RANGEM()   2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C               FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C               RANGEM(2,L) IS THE UPPER BOUND
C  UPPER      UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         RANGEM(1,L) = 0.0
         RANGEM(2,L) = 0.0
   50 CONTINUE

C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
            WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
         ENDIF

         IF (MPNT(L) .EQ. 1) THEN

C           THERE IS A POINT PRIOR ON M - THIS OVERRIDES ALL OTHER
C           INFORMATION:  ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR

            RANGEM(1,L) = MZERO(1,L)
            RANGEM(2,L) = 0.0

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS A PRIOR RANGE ON M, BUT NO Co CONSTRAINT USE Mo
```

```
            RANGEM(1,L) = MZERO(1,L)
            RANGEM(2,L) = MZERO(2,L)

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

       ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN

C         THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C         CONSTRAINT ADJUST THE LOWER BOUND OF Mo BY Mc

          LOWER = AMAX1(MZERO(1,L), MC(1,L))
          UPPER = MZERO(2,L)
          IF (UPPER .LT. LOWER) THEN
             WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
             CALL TRMNAT
          ELSE
             RANGEM(1,L) = LOWER
             RANGEM(2,L) = UPPER
          ENDIF

          IF (IOUT .EQ. 10) THEN
             WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                  ' MZERO(2,L) = ', MZERO(2,L)
             WRITE(8,*) 'MC(1,L) = ', MC(1,L)
             WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
             WRITE(8,*) 'RANEGM(1,L) = ', RANGEM(1,L),
     &                  ' RANGEM(2,L) = ', RANGEM(2,L)
          ENDIF

       ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN

C         THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO Co CONSTRAINT
C         INTERSECT THESE TWO RANGES

          LOWER = AMAX1(MZERO(1,L), MC(1,L))
          UPPER = AMIN1(MZERO(2,L), MC(2,L))
          IF (UPPER .LT. LOWER) THEN
             WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
             CALL TRMNAT
          ELSE
             RANGEM(1,L) = LOWER
             RANGEM(2,L) = UPPER
          ENDIF

          IF (IOUT .EQ. 10) THEN
             WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                  ' MZERO(2,L) = ', MZERO(2,L)
             WRITE(8,*) 'MC(1,L) = ', MC(1,L)
             WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
             WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                  ' RANGEM(2,L) = ', RANGEM(2,L)
          ENDIF

       ELSE

          WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
          CALL TRMNAT

       ENDIF

C      RESTRICT RANGE TO BE NON-NEGATIVE

       RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

       IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)
  100 CONTINUE
```

```
C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &        .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END


C***************************************************************************


C  SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C  DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  10FEB88     FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
     &                   MZERO, MPNT, MO, SIGMA2)

C  INPUTS:     RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
C              MO, SIGMA2
C  OUTPUTS:    RANGEM, MCHAT, MU, SIG, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

      REAL     MCHAT(2, MAXREG), MO(MAXREG), MU(MAXREG),
     &         MZERO(2, MAXREG), RANGEM(2, MAXREG), SIG(MAXREG),
     &         SIGMA2(MAXREG)


C                      LIST OF VARIABLES
C
C  IOUT          OUTPUT DUMP CONTROLLER
C  L             CONTROLS DO LOOP FOR EACH REGION
C  LL            EQUAL TO NUMREG FOR A SET OF CALCULATIONS
C  MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C  MCHAT()       2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C                  C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
C                  MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND
C                  MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MO()          1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                  MEAN FOR EACH REGION
C  MPNT()        1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                  MZERO() FOR EACH REGION
C  MU()          1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                  DISTRIBUTION MEAN FOR EACH REGION
C  MZERO()       2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                  EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                  IS UPPER BOUND
C  NNODAT        Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NUMREG        NUMBER OF REGIONS OF INTEREST
C  RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                  FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
```

```
C                 RANGEM(2,L) IS THE UPPER BOUND
C  SIG()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C  SIGMA2()     1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 VARIANCE FOR EACH REGION

      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            MU(LL) = MO(LL)
            SIG(LL) = SQRT(SIGMA2(LL))
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
               WRITE(8,*) 'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
               WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
     &                     SIGMA2(LL)
            ENDIF

C           SPECIFY E(M) OF POSTERIOR FOR SAKE OF
C           CALCULATIONS IN SUBROUTINE EXPCTD

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
               MU(LL) = RANGEM(1,LL)
               SIG(LL) = 0.0
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL),
     &         ' MU = ', MU(LL), ' SIG = ', SIG(LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE

      RETURN
      END


C***********************************************************************


C  SUBROUTINE PAREST CONTROLS THE CALCULATIONS FOR THE PARAMETER
C  ESTIMATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  13FEB89       FORMAT/COMMENTS:  15SEP89
C     VERSION:  MATCHR V8.3, V8.4, V8.5 - FOR USE WITH PFM'S
C               MATGRM V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG,
     &                   ZROREG, RAND, NBND, STR, BIGK, BZERO, MM,
     &                   SBND)
```

```
C  INPUTS:    VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
C             NBND, STR
C  OUTPUTS:   BIGK, BZERO, MM, SBND
C  SUBPROGRAMS:  FINDM, FINDMN, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, VARY, ZROREG

      REAL     BIGK(0:MAXREG), BZERO, K, MEANZ, MM(0:MAXREG),
     &         MU(MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &         RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
     &         STR(MAXDAT, MAXREG), SZ2, ZZ(MAXDAT)

      DOUBLE PRECISION RAND


C                          LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C              EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         VALUE OF k - PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT    MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ     SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MU()      1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C              DISTRIBUTION MEAN FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NF()      2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C              SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP        TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()    1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C              SPECIFIC MATERIAL S/N DATA SET
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  RANGEM()  2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C              FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C              RANGEM(2,L) IS THE UPPER BOUND
C  RAND      RANDOM NUMBER SEED
C  SBND()    1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C              REGION CONTAINED IN NBND()
C  SIG()     1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
C              STANDARD DEVIATION FOR EACH REGION
C  STR()     2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C              DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SZ2       SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  VARY      CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO VARIATION;
C              1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;
C              3 - TRUNCATED NORMAL VARIATION
C  ZROREG    ZeRO REGion - VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C  ZZ()      1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C              Z = F(STR,NF,NBND,MM)

C  OBTAIN THE VALUES OF M FOR EACH REGION

      IF (VARY .LE. 2) THEN

C        UNIFORM OR NO VARIATION IN M IS DESIRED

         CALL FINDM (RAND, NUMREG, RANGEM, MM)

      ELSE
```

```
C         NORMAL VARIATION IN M IS DESIRED
          CALL FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
      ENDIF
C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)
      CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)
C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)
      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)
C  CALCULATE THE VALUES FOR k AND BETAo FROM THE SAMPLE MEAN
C  AND VARIANCE
      CALL KBETA (MEANZ, SZ2, K, BZERO)
C  CALCULATE THE VALUE OF K FOR EACH REGION WHERE A = K ** M
      CALL FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)
C  CALCULATE STRESS TIE-POINTS
      CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)
C  WRITE RESULTS TO FILE
C      WRITE(7,900) NUMREG, BZERO
C      DO 200 L = ZROREG, NUMREG
C         WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
C  200 CONTINUE
C      WRITE(7,920)
C  FORMAT STATEMENTS
  900 FORMAT(///,2X,'SELECTED VALUES OF S/N CURVE PARAMETERS',
     &         //,2X,'NUMBER OF REGIONS: ',I4,5X,'BETAo = ',F8.4,
     &         //,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',5X,
     &         'STRESS BOUND',/)
  910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E9.3,6X,E11.5)
  920 FORMAT(///)

      RETURN
      END


C*************************************************************************


C  SUBROUTINE FINDM CALCULATES THE VALUE OF M FOR EACH REGION BY
C  SAMPLING OFF THE APPROPRIATE M RANGE
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

C  INPUTS:   RAND, NUMREG, RANGEM
C  OUTPUTS:  MM
C  SUBPROGRAMS:  RANDOM, TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)
```

```
      COMMON  IOUT
      INTEGER IOUT, L, NUMREG
      REAL    MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X
      DOUBLE PRECISION RAND

C                        LIST OF VARIABLES
C
C  IOUT        OUTPUT DUMP CONTROLLER
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()        1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  PICK()      1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C  RAND        RANDOM NUMBER SEED
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C  X           UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                OFF THE RANGE ON M

C  INITIALIZE MM()

      DO 50 L = 0, MAXREG
         MM(MAXREG) = 0.0
   50 CONTINUE

C  BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG

         PICK(1) = 0.0
         PICK(2) = 0.0

         IF (RANGEM(2,L) .EQ. 0.0) THEN
C           M IS SPECIFIED AS A POINT VALUE
            MM(L) = RANGEM(1,L)
            IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                                   ' MM(L) =', MM(L)
         ELSEIF (L .EQ. 1) THEN
C           SAMPLE ON EXISTING RANGE
            CALL RANDOM(X, RAND)
            MM(L) = (RANGEM(2,L) - RANGEM(1,L)) * X + RANGEM(1,L)
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                    ' RANGEM(2,L) =', RANGEM(2,L)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS - THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              SAMPLE ON ADJUSTED RANGE
               CALL RANDOM (X, RAND)
               MM(L) = (PICK(2) - PICK(1)) * X + PICK(1)
            ENDIF
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'L =', L, ' MM(L-1) =', MM(L-1),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
```

```
         ENDIF
  100 CONTINUE
      RETURN
      END


C**********************************************************************


C*********************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C*********************************************************************

      SUBROUTINE RANDOM (FRAC, RAND)
C     IMPLICIT NONE
      COMMON  IOUT
      INTEGER IOUT
      REAL    FRAC
      DOUBLE PRECISION  RANA, RANC, RAND, RANDIV, RANM, RANSUB,
     &        RANT, RANX

C               LIST OF VARIABLES
C
C  FRAC       UNIFORM (0,1) RANDOM VARIATE
C  IOUT       OUTPUT DUMP CONTROLLER
C  RANA       CONSTANT FOR LCG
C  RANC       CONSTANT FOR LCG
C  RAND       RANDOM NUMBER SEED
C  RANDIV     INTERNAL CALCULATION
C  RANM       CONSTANT FOR LCG
C  RANSUB     INTERNAL CALCULATION
C  RANT       INTERNAL CALCULATION
C  RANX       INTERNAL CALCULATION


C          USING LCG RANDOM # GENERATOR

      RANA = 671093.0
      RANC = 7090885.0
      RANM = 33554432.0

   10 RANX = RANA * RAND + RANC
      RANDIV = RANX / RANM
      RANT = DINT(RANDIV)
      RANSUB = RANT * RANM
      RAND = RANX - RANSUB
      FRAC = SNGL(RAND / RANM)

      IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
      IF (IOUT .EQ. 2) WRITE(8,*)'RANX =', RANX, ' RANDIV =', RANDIV,
     &   ' RANT =', RANT, ' RANSUB =', RANSUB, ' RAND =', RAND,
     &   ' FRAC =', FRAC

      RETURN
      END
```

```
C     NOTES:  IOUT=2 DUMPS TO SCREEN

C***********************************************************************

C  SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C  SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13FEB89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)

C  INPUTS:   RAND, NUMREG, MU, SIG, RANGEM
C  OUTPUTS:  MM
C  SUBPROGRAMS:  NORMGN, TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL     MM(0:MAXREG), MU(MAXREG), PICK(2), RANGEM(2, MAXREG),
     &         SIG(MAXREG), X

      DOUBLE PRECISION RAND

C                          LIST OF VARIABLES
C
C  IOUT        OUTPUT DUMP CONTROLLER
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()        1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MU()        1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  PICK()      1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C  RAND        RANDOM NUMBER SEED
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C  SIG()       1-D ARRAY CONTAINING THE STANDARD DEVIATION OF M FOR EACH
C                REGION
C  X           NORMAL(MU,SIGMA) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                OFF THE RANGE ON M

C  INITIALIZE MM()

      DO 50 L = 0, MAXREG
         MM(MAXREG) = 0.0
   50 CONTINUE

C  BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG

         PICK(1) = 0.0
         PICK(2) = 0.0

         IF (RANGEM(2,L) .EQ. 0.0) THEN
C           M IS SPECIFIED AS A POINT VALUE
            MM(L) = RANGEM(1,L)
            IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                                   ' MM(L) =', MM(L)
         ELSEIF (L .EQ. 1) THEN
```

```
C          SAMPLE ON EXISTING RANGE
   10      CALL NORMGN (RAND, MU(L), SIG(L), X)
           IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
           MM(L) = X
           IF (IOUT .EQ. 10) THEN
              WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                   ' RANGEM(2,L) =', RANGEM(2,L)
              WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
           ENDIF
        ELSE
C          ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C          AND THEN SAMPLE
           PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
           PICK(2) = RANGEM(2,L)
           IF (PICK(1) .GT. PICK(2)) THEN
C             NO RANGE EXISTS - THIS SHOULD NOT BE POSSIBLE
C             STOP PROGRAM
              WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
              CALL TRMNAT
           ELSE
C             SAMPLE ON ADJUSTED RANGE
   20         CALL NORMGN (RAND, MU(L), SIG(L), X)
              IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
              MM(L) = X
           ENDIF
           IF (IOUT .EQ. 10) THEN
              WRITE(8,*) 'L =', L, ' MM(L-1) =', MM(L-1),
     &                   ' RANGEM(1,L) =', RANGEM(1,L)
              WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
              WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                   ' MM(L) =', MM(L)
           ENDIF
        ENDIF

  100 CONTINUE

      RETURN
      END


C***********************************************************************


C***********************************************************************
C  SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C  WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  3FEB88
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
C
C  The random variates are generated using the "Direct Method"
C  Abramowitz, M., and Stegun, I. A., editors, Handbook of
C    Mathematical Functions, National Bureau of Standards, Applied
C    Mathematics Series 55, Issued June 1964, Ninth Printing, November
C    1970 with corrections, pg. 953.
C***********************************************************************

      SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C     SUBPROGRAM:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2

      PARAMETER (PI = 3.1415926536)

      INTEGER IOUT
```

```
C              LIST OF VARIABLES
C
C   FRAC      UNIFORM(0,1) RANDOM VARIATE
C   IOUT      OUTPUT DUMP CONTROLLER
C   MU        MEAN OF NORMAL DISTRIBUTION
C   RAND      RANDOM NUMBER SEED
C   SIGMA     STANDARD DEVIATION OF NORMAL DISTRIBUTION
C   X         NORMAL RANDOM VARIATE
C   U1        UNIFORM RANDOM NUMBER U(0,1)
C   U2        UNIFORM RANDOM NUMBER U(0,1)
C   Z1        NORMAL RANDOM NUMBER ON N(0,1)
C   Z2        NORMAL RANDOM NUMBER ON N(0,1)

      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'RAND =', RAND, ' MU =', MU, ' SIGMA =', SIGMA

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

      CALL RANDOM (FRAC, RAND)
      U2 = FRAC
      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'U1 =', U1,' U2 =', U2

      Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
      Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)

      X = SIGMA * Z1 + MU
      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'Z1 =', Z1, ' Z2 =', Z2, ' X =', X

      RETURN
      END


C***********************************************************************


C  SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM
C  THE S/N DATA INTO THE VARIABLE Z = Ln(X)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C  INPUTS:   NPTS, STR, NF, NUMREG, MM, NBND
C  OUTPUTS:  NP, ZZ

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON IOUT

      INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG

      REAL      MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &          STR(MAXDAT, MAXREG), ZZ(MAXDAT)

C                  LIST OF VARIABLES
C
C  I           CONTROLS DO LOOP FOR EACH DATA POINT
C  IOUT        OUTPUT DUMP CONTROLLER
C  K           CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
C  L           CONTROLS DO LOOP FOR EACH REGION
```

```
C  LL          CONTROLS INNER DO LOOP FOR EACH REGION
C  MAXDAT      MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()        1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
C  MML         EQUAL TO MM(L) FOR A SET OF CALCULATIONS
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()      1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C                SPECIFIC MATERIAL S/N DATA SET
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  ZZ()        1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
C                Z = F(STR,NF,NBND,MM)


C  INITIALIZE VARIABLES

      NP = 0

      DO 50 I = 1, MAXDAT
         ZZ(I) = 0.0
   50 CONTINUE


C  BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG
         MML = MM(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =',
     &        MML, ' NPTS =', NPTS(L)

         DO 200 K = 1, NPTS(L)
            NP = NP + 1
            ZZ(NP) = ALOG(STR(K,L)) + ALOG(NF(K,L)) * (1.0 / MML)
            IF (IOUT .EQ. 10) WRITE(8,*)'K =', K, ' NP =', NP, ' NF =',
     &         NF(K,L), ' STR =', STR(K,L), ' ZZ =', ZZ(NP)

            DO 300 LL = 2, L
               ZZ(NP) = ZZ(NP) + ALOG(NBND(LL-1))
     &                  * ((1.0 / MM(LL-1)) - (1.0 / MM(LL)))
               IF (IOUT .EQ. 10) WRITE(8,*)'LL =', LL, ' NBND(LL-1) =',
     &            NBND(LL-1), ' MM(LL-1) =', MM(LL-1), ' MM(LL) =',
     &            MM(LL), ' ZZ =', ZZ(NP)
  300       CONTINUE

  200    CONTINUE

  100 CONTINUE


      RETURN
      END


C***********************************************************************


C  SUBROUTINE SMNVAR CALCULATES THE Sample MeaN and VARiance OF
C  Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  24AUG87     COMMENTS:  13JUL89
C     VERSION:  MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V3.3, V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE SMNVAR (NP, ZZ, MEANZ, SZ2)

C  INPUTS:   NP, ZZ
```

```
C  OUTPUTS:  MEANZ, SZ2
C      IMPLICIT NONE
       INTEGER MAXDAT
       PARAMETER (MAXDAT = 50)
       COMMON IOUT
       INTEGER I, IOUT, NP
       REAL    MEANZ, SZ2, ZZ(MAXDAT)

C                    LIST OF VARIABLES
C
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C               DATA SET
C  SZ2        SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  ZZ()       1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C               Z = F(STR,NF,NBND,MM)

C  INITIALIZE VARIABLES
      MEANZ = 0.0
      SZ2 = 0.0

C  CALCULATE THE MEAN OF ZZ(), MEANZ
      DO 100 I = 1, NP
         MEANZ = MEANZ + ZZ(I)
         IF (IOUT .EQ. 10) WRITE(8,*)'NP =', NP, ' I =', I,
     &          ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
  100 CONTINUE
      MEANZ = MEANZ / FLOAT(NP)
      IF (IOUT .EQ. 10) WRITE(8,*)' MEANZ =', MEANZ

C  CALCULATE THE VARIANCE OF ZZ(), SZ2
      DO 200 I = 1, NP
         SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
         IF (IOUT .EQ. 10) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
  200 CONTINUE
      SZ2 = SZ2 / FLOAT(NP - 1)
      IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 =', SZ2

      RETURN
      END


C*************************************************************************


C  SUBROUTINE KBETA CALCULATES k AND BETAo FROM THE SAMPLE MEAN AND
C  VARIANCE OF Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)

C  INPUTS:    MEANZ, SZ2
C  OUTPUTS:   K, BZERO
```

```
C     IMPLICIT NONE

      REAL    PI

      PARAMETER (PI = 3.1415926536)

      COMMON  IOUT

      INTEGER IOUT

      REAL    BZERO, K, MEANZ, SZ, SZ2


C                 LIST OF VARIABLES
C
C  BZERO        VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE
C                 SPECIFIC MATERIAL S/N DATA SET
C  IOUT         OUTPUT DUMP CONTROLLER
C  K            VALUE OF k - PARAMETER CHARACTERIZING SPECIFIC MATERIAL
C                 DATA BASE
C  MEANZ        SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  PI           SELF EXPLANATORY CONSTANT
C  SZ           SZ2 ** 0.5
C  SZ2          SAMPLE VARIANCE OF THE TRANSFORMED DATA,
C                 Z = F(STR, NF, NBND, MM)


C  PERFORM CALCULATIONS

      SZ = SZ2 ** 0.5

      BZERO = PI / (SZ * (6.0 ** 0.5))

      K = MEANZ

C  DATA DUMP STATEMENTS

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'SZ2 =', SZ2, ' SZ =', SZ
         WRITE(8,*) 'MEANZ =', MEANZ, ' K = ', K, ' BZERO =', BZERO
      ENDIF

      RETURN
      END


C************************************************************************


C  SUBROUTINE FINDK CALCULATES THE VALUE OF K, WHERE A = K ** M FOR
C  EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  7JUN88
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C  INPUTS:   BZERO, K, MM, NBND, NUMREG
C  OUTPUTS:  BIGK

C     IMPLICIT NONE

      INTEGER MAXREG

      REAL    GAMMA

      PARAMETER (GAMMA = 0.57721566490, MAXREG = 3)
```

```
      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    BIGK(0:MAXREG), BZERO, K, MM(0:MAXREG), NBND(0:MAXREG)

C                     LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC
C              MATERIAL DATA BASE
C  GAMMA     EULER'S CONSTANT
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         VALUE OF k - PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C              DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST

C  INITIALIZE VARIABLES

      DO 50 L = 0, MAXREG
         BIGK(L) = 0.0
   50 CONTINUE

C  CALCULATE K FOR REGION ONE

      BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
C      WRITE(7,*) 'REGION:  1,  K =', BIGK(1)
      IF (IOUT .EQ. 10) WRITE(8,*)'BZERO =', BZERO, ' k =', K,
     &   ' GAMMA =', GAMMA, ' BIGK(1) =', BIGK(1)

C  CALCULATE K FOR REMAINING REGIONS

      DO 100 L = 2, NUMREG
         BIGK(L) = BIGK(L-1) * NBND(L-1)
     &            ** ((1.0 / MM(L)) - (1.0 / MM(L-1)))
C         WRITE(7,*) 'REGION ', L, ' K =', BIGK(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' NBND(L-1) =',
     &      NBND(L-1), ' MM(L) =', MM(L), ' MM(L-1) =', MM(L-1),
     &      ' BIGK(L) =', BIGK(L)
  100 CONTINUE

      RETURN
      END


C************************************************************************


C  SUBROUTINE FINDSB CALCULATES THE REGION 'TIE-POINTS' - THE STRESS
C  VALUES WHICH CORRESPOND TO THE "LIFE BOUNDARIES" ACCORDING TO THE
C  RANDOMLY SELECTED Ms, AND THE Ks CALCULATED FROM THE BETA AND k
C  CHARACTERIZING SPECIFIC MATERIAL
C  PROGRAMMER:  L. NEWLIN
C        DATE:  22DEC88
C     VERSION:  MATCHR V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C  INPUTS:   NUMREG, ZROREG, NBND, BIGK, MM
C  OUTPUTS:  SBND

C     IMPLICIT NONE

      INTEGER MAXREG
```

```
      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG, ZROREG

      REAL    BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
     &        SBND(0:MAXREG)

C             LIST OF VARIABLES
C
C  BIGK()   1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C             FOR EACH REGION
C  IOUT     OUTPUT DUMP CONTROLLER
C  L        CONTROLS DO LOOP FOR EACH REGION
C  MAXREG   MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()     1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()   1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C             REGIONS OF INTEREST
C  NUMREG   NUMBER OF REGIONS OF INTEREST
C  SBND()   1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C             CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C             REGION CONTAINED IN NBND()
C  ZROREG   ZeRO REGion — VALUES CHOSEN TO FACILITATE REGION DO LOOP
C             BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO REGION

C  INITIALIZE SBND()

      DO 50 L = 0, MAXREG
         SBND(L) = 0.0
   50 CONTINUE

C   CALCULATE SBND(0) IF ZROREG = 0

      IF (ZROREG .EQ. 0) THEN
         SBND(0) = BIGK(1) * NBND(0) ** (-1.0 / MM(1))
      ENDIF

C  CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES

      DO 100 L = 1, NUMREG
         IF (NBND(L) .GE. 1.0E+36) THEN
            SBND(L) = 0.0
         ELSE
            SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
         ENDIF
  100 CONTINUE

      RETURN
      END


C************************************************************************


C  THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
C  MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C  TRANSFORM METHOD"
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  18MAR87      COMMENTS:  15SEP89
C     VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                      V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                      V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
```

```
      SUBROUTINE WEIBGN (BETA, RAND, WEIB)
C  INPUTS:   BETA, RAND
C  OUTPUTS:  WEIB
C  SUBPROGRAMS:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER IOUT

      REAL    ARG, BETA, ETA, FRAC, WEIB

      DOUBLE PRECISION RAND

C                 LIST OF VARIABLES
C
C  ARG      INTERMEDIATE CALCULATION VARIABLE
C  BETA     WEIBULL DISTRIBUTION SHAPE PARAMETER
C  ETA      WEIBULL DISTRIBUTION LOCATION PARAMETER
C  FRAC     UNIFORM (0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  RAND     RANDOM NUMBER SEED
C  WEIB     WEIBULL(BETA,ETA) GENERATED RANDOM VARIATE


C     CALCULATE CONSTRAINED ETA

      ETA = 1.0 / (ALOG(2.0) ** (1.0 / BETA))

C     GENERATE WEIBULL RANDOM VARIATE

      CALL RANDOM(FRAC, RAND)
      ARG = -ALOG(1.0 - FRAC)
      WEIB = ETA * ARG**(1.0/BETA)
      IF (IOUT .EQ. 10) WRITE(8,*)'BETA = ', BETA, ' ETA =', ETA,
     &    ' FRAC =', FRAC, ' ARG =', ARG, ' WEIB =', WEIB

      RETURN
      END


C***********************************************************************


C  SUBROUTINE KOMO CALCULATES Ko AND Mo FOR THE ZERO REGION (NO DATA
C  REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT
C  AT SZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
C  PROGRAMMER :  L. NEWLIN
C        DATE:   1AUG91
C     VERSION:   MATCHR V8.5   MATGRM V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
     &                 FACTR, NUMREG)

C  INPUTS:   SZERO, BIGK, MM, NBND, TRSBND, FACTR
C  OUTPUTS:  TRBIGK, MM, TRSBND

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT
```

```
      INTEGER   IOUT, L, NUMREG

      REAL      BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1          SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)

C                        LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  FACTR      SCALE FACTOR = PHI * KRATIO * Z
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NUMREG     NUMBER OF REGIONS
C  SCLK       ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
C  SZERO      STRESS TENSILE TEST POINT, So
C  TRBIGK()   1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP
C               SBND(0) < So FOR EACH TRIAL
C  TRSBND()   1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE
C               LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()
C               ADJUSTED BY VARIATION PARAMETERS FOR EACH TRIAL


      BIGK(0) = SZERO

      IF (TRSBND(0) .GT. SZERO) THEN
         SCLK = SZERO/TRSBND(0)
         DO 100 L = 0, NUMREG
            TRBIGK(L) = BIGK(L) * SCLK
            TRSBND(L) = TRSBND(L) * SCLK
 100     CONTINUE
      ELSE
         TRBIGK(0) = SZERO/FACTR
         MM(0) = MM(1) * ((ALOG (BIGK(1)) - ALOG (TRSBND(0))
     &            + ALOG (FACTR)) / (ALOG (SZERO) - ALOG (TRSBND(0))))
      ENDIF
C
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'SZERO = ', SZERO, ' BIGKo = ', TRBIGK(0)
         WRITE(8,*) 'FACTOR = ', FACTR, ' BIGK1 = ', TRBIGK(1)
         WRITE(8,*) 'MM1 = ', MM(1), ' MMo = ', MM(0)
      ENDIF


      RETURN
      END

C*************************************************************************


C  FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
C  BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB89
C     VERSION:  MATCHR V8.3, V8.4, V8.5 - FOR USE WITH PFM'S
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      REAL FUNCTION GTLIFE (S, MM, LNA, LPHIM, KRATIO, LNZ, SBND,
     &                      ZROREG, NUMREG, SZERO)

C  INPUTS:   S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO
C  OUTPUTS:  GTLIFE

C     IMPLICIT NONE
```

```
      INTEGER IOUT, L, MAXREG, NUMREG, ZROREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      REAL     GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
     &         MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

C                          LIST OF VARIABLES
C
C   GETLIF     VALUE TO BE ASSIGNED TO GTLIFE — CYCLES TO FAILURE FOR
C                THE REQUIRED STRESS LEVEL
C   IOUT       OUTPUT DUMP CONTROLLER
C   KRATIO     RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C   L          CONTROLS DO LOOP FOR EACH REGION
C   LNA()      1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C   LNZ        NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C   LPHIM()    1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
C                PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C   MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C   MM()       1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C   NUMREG     NUMBER OF REGIONS OF INTEREST
C   S          VALUE OF STRESS (PSI) FOR WHICH A VALUE OF LIFE (CYCLES TO
C                FAILURE) IS REQUIRED
C   SBND()     1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C                CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
C                CONTAINED IN NBND()
C   SZERO      STRESS TENSILE POINT, So
C   TEMP       TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER
C                FLOWS
C   ZROREG     ZeRO REGion — VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO REGION

      GETLIF = 0.0

C  CALCULATE CYCLES TO FAILURE

      IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
         GETLIF = 1.0
      ELSE
         DO 100 L = ZROREG, NUMREG
            IF (S .GT. SBND(L)) THEN
               TEMP = LNA(L) + LPHIM(L) + MM(L) * ( - ALOG(S)
     &                    +  ALOG (KRATIO) + LNZ)
               IF (TEMP .GT. 86.0) THEN
                  TEMP = 86.0
               ENDIF
               GETLIF = EXP (TEMP)
               GOTO 150
            ENDIF
 100     CONTINUE
      ENDIF
 150  CONTINUE

      GTLIFE = GETLIF

      RETURN
      END


C***********************************************************************


C  SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C  M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB88
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
```

```
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE SORTM (ALLM, NUMREG, NUM)

C   INPUTS:  ALLM, NUMREG, NUM
C  OUTPUTS:  ALLM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER I, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG

      PARAMETER (MAXMM = 20001, MAXREG = 3)

      LOGICAL INORDR

      REAL    ALLM(MAXMM, MAXREG), TEMP

C                    LIST OF VARIABLES
C
C  ALLM()     2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
C  I          CONTROLS INSERTION POINTER
C  INC        SORT INCREMENT VARIABLE
C  INORDR     FLAG TO INDICATE WHETHER SORT IS FINISHED
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXMM      MAXIMUM NUMBER OF M'S TO BE SORTED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUM        NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  TEMP       TEMPORARY SORTING VARIABLE


      DO 400 L = 1, NUMREG

    5    INC = NUM
   10    IF (INC .GT. 1) THEN
            INC = INC / 2
   20       INORDR = .TRUE.

            DO 300 I = 1, (NUM - INC)
               IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN
                  TEMP = ALLM(I,L)
                  ALLM(I,L) = ALLM(I + INC, L)
                  ALLM(I + INC, L) = TEMP
                  INORDR = .FALSE.
               ENDIF
  300       CONTINUE

            IF (.NOT. INORDR) GOTO 20
            GOTO 10
         ENDIF

  400 CONTINUE

      RETURN
      END
```

Reference

[1] Moore, N., et al., An Improved Approach for Flight Readiness Certification – Methodology for Failure Risk Assessment and Application Examples, JPL Publication 92-15, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, June 1, 1992.